Linux härten

Viele Distributionen legen mehr Wert auf Flexibilität und Bedienbarkeit als auf die Systemsicherheit. Eine Installation dagegen für eine bestimmte Aufgabe maßzuschneidern und abzusichern, heißt „Härten“. Dabei helfen Tools wie Bastille, Security Blanket, Debian Harden oder das Yast-Modul Permissions. Ralf Spenneberg

Linux läuft fast überall, von Embedded Systems und Firewalls über Desktop-Systeme bis hin zu Servern. Zwar gibt es für viele Aufgaben optimierte Distributionen, viele Anwender setzen aber einheitlich auf Open Suse, Fedora, Debian oder Ubuntu. Wie ihre kommerziellen Schwestern sind diese Linux-Varianten universell ausgelegt, sodass ihre Sicherheit gelegentlich hinter der Bedienungsfreundlichkeit zurücksteht.

Einen dicken Vorteil haben die großen Distributionen aber: Es gibt sehr zuverlässig und schnell Sicherheitsupdates. Daher lohnt es sich oft, diese zu nehmen und auf die eigenen Bedürfnisse anzupassen, sprich sie zu härten. Speziell für den Betrieb von Firewalls oder Servern gibt es viele Anleitungen und fertige Tools. Leider gibt es keinen Standard, der allgemeingültig beschreiben würde, welche Schritte beim Härten nötig sind. Die folgenden gehören fast immer dazu:

- Überflüssige Dienste deaktivieren
- Überflüssige Softwarepakete deinstallieren
- Alle verfügbaren Updates einspielen
- Bootloader und BIOS mit Kennwörtern schützen
- Zugriff vom Netzwerk einschränken
- SSH sicher konfigurieren
- Kernel sicher konfigurieren
- Zugriffsrechte auf sensible Dateien sicher vergeben
- Nicht dringend nötige Set-UID- und Set-GID-Rechte entfernen
- Überflüssige Benutzerkonten deaktivieren
- Kennwortrichtlinien verwenden (Alter, Güte, Wiederverwendung)
- Konten nach mehreren fehlerhaften Anmeldungen automatisch sperren
- Login-Banner anzeigen
- Systemprotokollierung konfigurieren
- Protokolle auf einem zentralen Log-Server speichern
- Die eingebaute Firewall konfigurieren

Sicherlich gibt es bei einigen Punkten unterschiedliche Ansichten, schließlich passt nicht jeder Aspekt zu jedem System. Der Aufwand ist aber immer hoch, alles von Hand abzuarbeiten daher mühsam und fehleranfällig. Zudem empfiehlt es sich, vergleichbare Installationen nachvollziehbar und gleichartig zu härten, idealerweise per Skript. Statt solche Skripte selbst zu schreiben, bietet es sich an, fertige Skripte, die zur eigenen Distribution gehören (etwa bei Debian und Suse), oder allgemeine Werkzeuge zu verwenden, beispielsweise Bastille (1) oder Security Blanket (2).

Debian

Für Debian existiert ein umfangreiches Howto (3), das die nötigen Schritte auf Englisch, Deutsch und Französisch beschreibt. Zusätzlich stellt die Distribution etliche Pakete bereit, die verhindern, dass der Admin irrtümlich gefährliche Software installiert oder Sicherheitssoft-

Tabelle 1: Debian Harden

Paket	Aufgabe
harden-doc	Dokumente zu Sicherheitsthemen, derzeit nur das „Securing Debian Manual"
harden-development	Werkzeuge zum Entwickeln sicherer Programme
harden	Installiert die nachfolgenden virtuellen Pakete:
harden-clients	Entfernt alle Clients, die als unsicher gelten
harden-environment	Soll künftig helfen, eine sichere Umgebung zu konfigurieren. Enthält derzeit Pakete zur lokalen Intrusion Detection (IDS)
harden-nids	Richtet ein NIDS ein (Network Intrusion Detection System)
harden-remoteaudit	Bindet Pakete ein, die Systeme aus der Ferne prüfen
harden-servers	Entfernt alle Server, die als unsicher gelten
harden-surveillance	Bindet Pakete ein, die lokale Dienste und das Netzwerk überwachen
harden-tools	Bindet Pakete ein, die helfen, das System abzusichern (Integritätsprüfung, Intrusion Detection, Kernel-Patches uvm.)

ware versehentlich entfernt. Der Trick: Die in **Tabelle 1** genannten virtuellen Pakete sorgen durch ihre Abhängigkeiten dafür, dass keine kritische Software auf das System kommt oder wichtige wieder verschwindet.

Während »harden-nids« und »harden-tools« Software installieren, die das System sicherer macht, verhindern »harden-clients« und »harden-servers« die Installation von Paketen, die als unsicher gelten. **Listing 1** zeigt das anhand des Telnet-Pakets: Versucht der Admin den Telnet-Daemon zu installieren, dann erkennt das Paketverwaltungssystem, dass sich Telnetd und Harden-Servers gegenseitig ausschließen. Das geniale an diesem Ansatz: Egal, welches Frontend zur Paketverwaltung der Admin verwendet, Debian warnt ihn vor solch heiklen Schritten. Die Konfiguration oder die Rechte des Systems modifiziert jedoch keines der Harden-Pakete.

Update-Falle

So wichtig Updates sind: Auf gehärteten Systemen stellen sie eine Herausforderung dar. Hat der Admin Dateirechte und Konfigurationen angepasst, dann setzt ein Update-Paket die Korrekturen möglicherweise wieder auf unsichere Standardwerte zurück. Das Härten ist daher kein einmaliger Vorgang, es gilt bei jeder Änderung und jedem Update den aktuellen Status zu prüfen und bei Bedarf einzelne Schritte zu wiederholen.

Open Suse

Bei Open Suse ist Yast die zentrale Schaltstelle für Systemeinstellungen. Der Befehl »yast security« lässt allgemein zwischen dem Einsatzgebiet Heimarbeitsplatz, vernetzter Arbeitsplatzrechner und Server wählen. Zusätzlich erlaubt es Yast, die Kennwörter zu konfigurieren sowie spezielle Admin-Rechte zu vergeben, etwa um den Rechner herunterzufahren. Für sichere Dateiberechtigungen gibt es ein eigenes Modul (**Abbildung 1**).

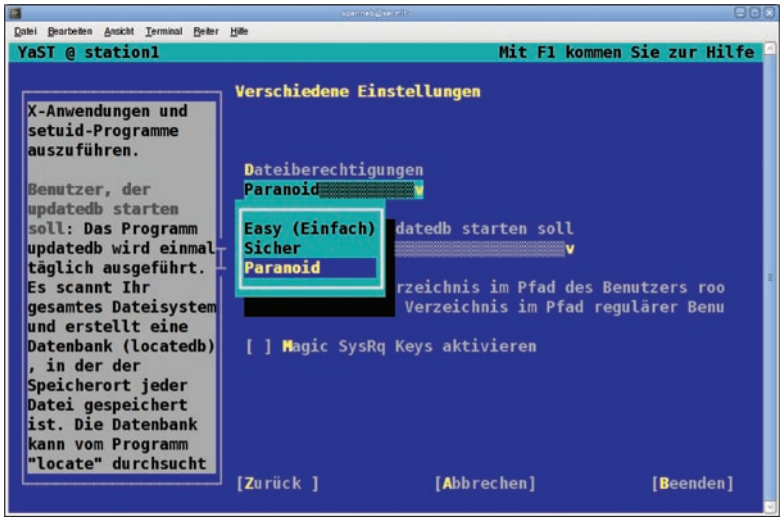

Abbildung 1: Suses Yast – hier in der Konsolen-Variante – gibt dem Admin mehrere Sicherheitsstufen zur Auswahl. Bei »Paranoid« dürfen normale Benutzer kaum noch etwas anstellen.

Listing 1: Harden-Warnung

```
01 # apt-get install telnetd
02    Die folgenden Pakete werden
   ENTFERNT:
03      harden-servers
04    Die folgenden NEUEN Pakete werden
   installiert:
05      telnetd
06    Möchten Sie fortfahren? [J/n]
```

Yast greift auf die Einstellungen in den vier Konfigurationsdateien »/etc/permissions.[local|easy|secure|paranoid]« zurück. Bei jedem Start von Suseconfig prüft und korrigiert das Tool die Dateirechte anhand dieser Vorgaben. Dabei setzt es auch unsichere Änderungen wieder zurück, die Benutzer irrtümlich vorgenommen haben. Wünscht der Admin bei Abweichungen lediglich eine Warnung, kann er dies in »/etc/sysconfig/security« einstellen.

Bastille

Der Klassiker unter den Härtungsskripten ist Bastille Linux, das neuerdings Bastille Unix heißt **(1)**. Das Open-Source-Tool härtet Red Hat, Fedora, Open Suse, Mandriva, Debian, Gentoo, HP-UX und Mac OS X – für diese Systeme stehen fertige Pakete zum Download bereit. Entstanden ist Bastille unter der Leitung des Sicherheitsspezialisten Jay Beale und der Schirmherrschaft des renommierten Sans Institute. Leider startet die aktuelle Version 3.2.1 auf dem Testsystem nicht – vermutlich, weil es auf dem 32-Bit-Rechner alle Bibliotheken unter »/usr/lib64« deponiert. Die Vorgängerfassung 3.0.9 funktioniert hingegen **(4)**. Aber auch die älteren Versionen sind nicht ohne Fehler. So haben sie häufig Probleme aktuelle Linux-Distributionen zu erkennen.

Bastille-Linux ist in Perl programmiert und benötigt für die Anzeige seiner Oberfläche entweder Perl-TK (GUI) oder Perl-Curses (Text-Interface). Anschließend starten »bastille -x« die GUI-Version (**Abbildung 2**) und »bastille -c« die Textvariante (**Abbildung 3**)

Bastille schlägt anschließend viele kleine Änderungen in der Konfiguration des Systems vor. Der Admin entscheidet, welche davon er an-

wenden möchte. Dabei beschreibt Bastille die Problematik und stellt Vor- und Nachteile der Änderungen vor.

Ausreichend Geduld

Erfreulicherweise verändert Bastille das System nicht, während der Admin noch die Fragen beantwortet. Auch am Ende schreibt das Tool zunächst nur eine Konfigurationsdatei, die der Admin explizit auf dem System umsetzen muss (ein eigener Dialog fordert ihn dazu auf). Bastille unterstützt auch einen nicht interaktiven Modus »bastille -b«, in dem es seine Konfigurationsdatei einliest und das lokale System entsprechend anpasst. Damit löst es zwei häufige Aufgaben:

- Erneutes Härten nach Installation oder Update von Softwarepaketen
- Identisches Härten mehrerer gleichartiger Rechner

Seine Konfigurationsdatei sucht Bastille in »/etc/Bastille/«. Eine Default-Konfigurationsdatei liefern die Entwickler nicht mit, der Benutzer muss zunächst alle Fragen selbst beantworten. Beim ersten Mal dauert das etwa ein bis zwei Stunden.

Protokolliert

Beim Umsetzen der Einstellungen protokolliert Bastille jeden Schritt in »/var/log/Bastille/action-log«. In »/var/log/Bastille/error-log« informiert es über aufgetretene Fehler. Außerdem erzeugt es die Datei »/var/log/Bastille/TODO«, die weitere Maßnahmen beschreibt, die der Admin noch manuell erledigen sollte.

Möchte der Admin die Änderungen rückgängig machen, so bietet Bastille die Revert-Funktion »bastille -r«. Die stellt alle veränderten Einstellungen auf den Zustand vor dem Bastille-Aufruf zurück. Haben andere Programme oder der Admin selbst inzwischen weitere Modifikationen vorgenommen, dann nimmt Bastille eventuell auch diese wieder zurück. Wer vorab prüfen will, was Bastille zurücknimmt, wird in »/var/log/Bastille/revert/revert-actions« fündig.

Generell empfiehlt sich, alle Änderungen, die Bastille durchführt, zu prüfen. Wer nicht genau abschätzen kann, welche Auswirkungen eine Modifikation nach sich zieht läuft eventuell Gefahr, sein System in einen instabilen oder nicht funktionstüchtigen Zustand zu versetzen. Bastille hilft lediglich, an alle Möglichkeiten zu denken.

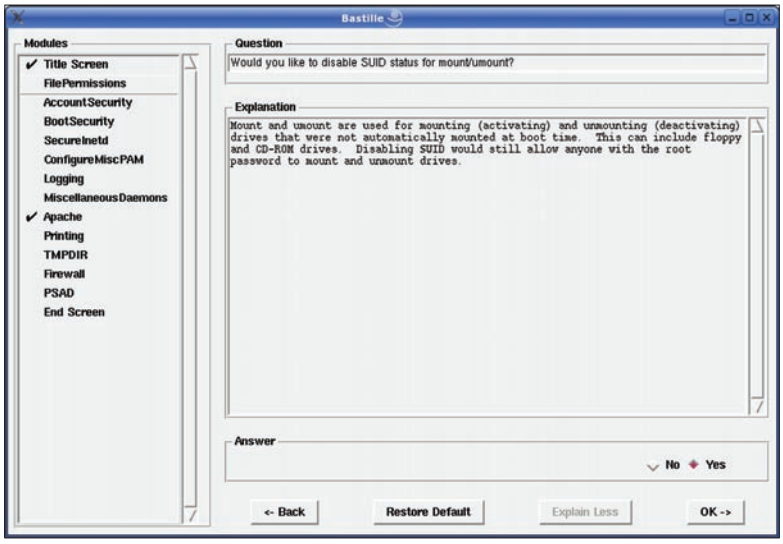

Abbildung 2: Am bequemsten ist die grafische Variante von Bastille zu bedienen. Sie fragt den Admin hier, ob er Set-UID-Rechte für Mount verwenden will und erklärt auch die Folgen.

► Security Blanket

Der kommerzielle Bastille-Konkurrent Security Blanket (3) wurde erstmals im September 2007 auf der Linux World Expo in San Francisco vorgestellt. Den Begriff „Security Blanket“ hat Charles M. Schulz im Peanuts-Comicstrip geprägt. Hier besitzt Linus ein Tuch (Blanket), das ihm eine gewisse Sicherheit gibt (Security). Die US-Firma TCS (Trusted Computer Solutions) hat bereits ähnliche Produkte für kommerzielle Unix-Varianten entwickelt und stellt mit Security Blanket (**Abbildung 4**) ein Werkzeug für die

Abbildung 3: Die alternative Textoberfläche von Bastille steht auch ohne X-Server zur Verfügung. Sie stellt die gleichen Fragen wie die GUI-Variante und liefert die gleichen Erklärungstexte.

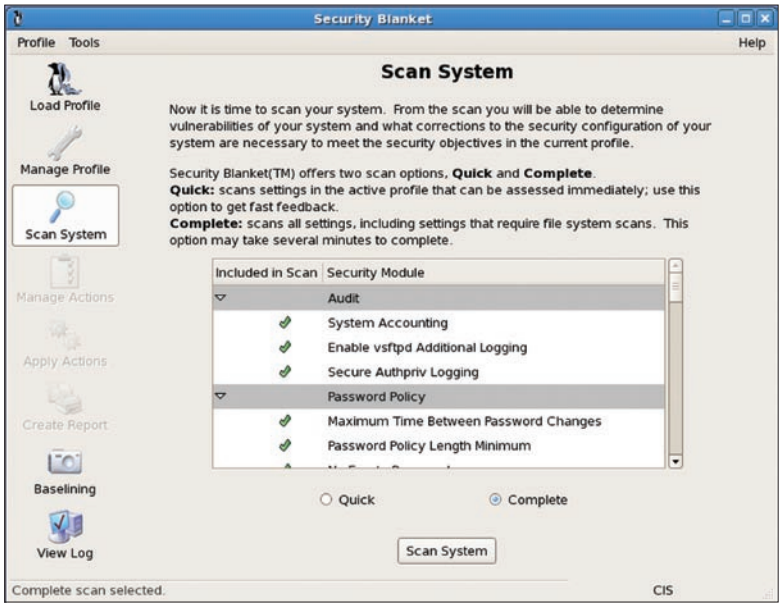

Abbildung 4: Das kommerzielle Produkt Security Blanket ist speziell auf RHEL ausgelegt und unterstützt nur dessen Softwarepakete. Immerhin ist das User Interface deutlich moderner als bei Bastille.

kommerziellen RHEL-Versionen 4 und 5 (Red Hat Enterprise Linux) und deren Ableger vor (etwa Cent OS). Eine Einzelplatzlizenz kostet knapp 200 US-Dollar.

Nützliche Ergänzung

Der Hersteller behauptet, wesentlich mehr Sicherheitsprobleme zu behandeln als Bastille. Dies ließ sich in einem Test allerdings nicht nachvollziehen – der Umfang ist ungefähr vergleichbar. Der Hersteller gibt auch an, dass Benutzer das Werkzeug ohne Unix-Kenntnisse einsetzen könnten. Auch diese Marketing-Behauptung ist nicht nachvollziehbar. Zwar hat TCS die Beschreibungstexte der Prüfungen und Änderungen viel weniger technisch formuliert, allerdings muss der Admin weiterhin abschätzen, ob eine Modifikation den Betrieb des Systems negativ beeinflusst. Ohne fundiertes Admin-Wissen ist seriöses Härten nicht möglich. Sowohl Bastille als auch Security Blanket haben noch eine interessante zusätzliche Funktion: Assessment und Reporting. Mit ihr prüft der Admin oder Auditor ein System auf sichere Konfiguration. Die Werkzeuge analysieren den aktuellen Status und schlagen anschließend Verbesserungen vor. Bastille erzeugt sogar einen HTML-Bericht. Diese Funktion eignet sich am Ende übrigens prima um eine sichere Systemkonfiguration in ansprechender Form zu dokumentieren.

Keines der vorgestellten Skripte entfernt überflüssige Komponenten. Die Tools können schließlich nicht wissen, welche Pakete der Admin bewusst installiert und welche er sich zufällig per Standardinstallation ins System geholt hat. Bleibt nur, mit »dpkg -l« oder »rpm -qa« die installierten Pakete zu listen und zu versuchen, alle eventuell überflüssigen Einträge zu entfernen. Im Zweifelsfall verhindert die Paketverwaltung dank der Paketabhängigkeiten eine Deinstallation wichtiger Komponenten.

Sinn und Unsinn

Moderne Distributionen enthalten MAC-Systeme (Mandatory Access Control) wie App Armor (Open Suse, Ubuntu, Mandriva) oder SE Linux (Debian, Fedora). Viele Admins mögen sich fragen, ob das klassische Härten dann noch nötig ist. Die klare Antwort: Ja. Im schlimmsten Fall deaktivieren ein findiger Angreifer oder der Admin selbst SE Linux und App Armor, dann ist das vermeintlich sichere System plötzlich ungeschützt. Außerdem sind die mitgelieferten Richtlinien oft recht locker gefasst, um die Funktionstüchtigkeit nicht einzuschränken. Sie erweitern also eher die Aufgabe des Härtens – auf manuelle Kontrolle der MAC-Regelsätze Für SELinux beschreibt ein eigener Artikel die Details. *(fjl)*

Der Autor

Ralf Spenneberg arbeitet als freier Unix/Linux-Trainer, Berater und Autor. Er veröffentlichte mehrere Bücher zu den Themen Intrusion Detection, Firewalling und Virtuelle Private Netzwerke. Vor wenigen Wochen ist sein neues Buch „SELinux & AppArmor" bei Addison Wesley erschienen.

Infos

(1) Bastille Linux: (http://www.bastille-unix.org)

(2) Security Blanket: (http://www.tcs-sec.com/SecurityBlanket.html)

(3) Securing Debian Manual: (http://www.debian.org/doc/manuals/securing-debian-howto/index.en.html); deutsche Übersetzung: (.../index.de.html)

(4) Bastille-Version 3.0.9: (http://prdownloads.sourceforge.net/bastille-linux/Bastille-3.0.9-1.0.noarch.rpm?download)