

Paketmanager für Linux

Antworten auf die häufigsten Fragen

Von Merlin Schumacher

Wofür Paketmanager?

? Warum gibt es überhaupt Paketmanager unter Linux? Windows braucht doch auch keinen!

! Paketmanager erleichtern die Installation und Verwaltung von Programmen und vor allem von Systembibliotheken erheblich. Unter Windows müssen alle für den Betrieb der installierten Programme notwendigen Bibliotheken in den passenden Versionen vorgehalten werden. Das bedeutet, dass der jeweilige Programm-Installer diese mitbringt und bei Bedarf installiert. Ein Paketmanager verwaltet die Bibliotheken auf Systemebene und braucht Pakete sonst nur einmal zu installieren. Zudem wissen die Paketmanager, welche Dateien zu den installierten Paketen gehören und können diese deshalb wieder vollständig deinstallieren.

Die beste Wahl

? Welcher ist der beste Paketmanager? Sollte ich bei der Wahl einer Linux-Distribution darauf achten, welchen Paketmanager sie verwendet?

! Moderne Paketmanager lassen aus Anwendersicht wenig zu wünschen übrig. Die Installation und Verwaltung von Software gelingt mit allen problemlos. Rpm kann darüber hinaus mit Hilfe von Prüfsummen die Integrität aller installierten Dateien verifizieren. Das ist vor allem im Enterprise-Bereich wichtig. Dpkg löst Änderungen von Abhängigkeiten auf, was vor allem beim Upgrade auf die neue Version einer Distribution wichtig ist.

Dank der Abstraktionsschicht PackageKit sind sogar distributionsübergreifende Frontends wie etwa Gnome Software möglich. So kann man auf jeder Distribution dieselben Oberflächen verwenden, selbst wenn im Hintergrund unterschiedliche Programme arbeiten. Einzig das Ubuntu Software Center sticht etwas heraus, weil

es neben „normalen“ Paketen auch Programme im Snap-Paketformat anbietet.

Flatpak, Snap & Co

? Werden Formate wie Flatpak und Snap die klassischen Paketmanager wie Dpkg oder Rpm ablösen?

! Vermutlich nicht. Zum einen liegen Flatpak und Snap die klassischen Paketmanager zugrunde, zum anderen ist das Bauen von Distributionen mit konventionellen Paketmanagern erheblich leichter. Ansätze wie Flatpak und Snap sollen die Bereitstellung von Software als installierbares und distributionsunabhängiges Paket erleichtern, da alle für den Betrieb notwendigen Bibliotheken mitgeliefert werden.

Zudem stellen Flatpak und Snap mehr oder weniger abgeschottete Laufzeitumgebungen für die Software dar, was bei klassischen Paketmanagern nicht der Fall ist. Mehr zu Flatpak und Snap finden Sie unter [1].

Externe Paketquellen

? Ich habe eine externe Paketquelle für ein Programm gefunden. Kann ich diese bedenkenlos in meinem System einbinden?

! Lassen Sie bei externen Paketquellen immer große Vorsicht walten! Durch eine externe Paketquelle installieren Sie Software auf Ihrem System, über deren Qualität Sie nichts wissen. Mit der Vertrauenswürdigkeit dieser Pakete steht und fällt auch die Sicherheit Ihres Systems. Achten Sie deshalb darauf, dass der Ersteller des Paketes vertrauenswürdig ist. Stammt es womöglich direkt vom Entwickler der Software? Wenn die Paketquelle vom Software-Anbieter selbst bereitgestellt wird, ist diese potenziell vertrauenswürdiger als ein Paket von einem unbekanntem Dritten. Wird die Paketquelle von vertrauenswürdigen Seiten empfohlen, wie etwa der offiziellen Dokumentation einer Distribution oder der Home-

page der Software, ist das Risiko bei der Installation geringer. Achten Sie bei der Einbindung externer Paketquellen unbedingt darauf, dass deren Pakete ebenso signiert sind wie die der offiziellen Paketquellen Ihrer Distribution.

Wechsel des Paketmanagers

? Kann ich den Paketmanager meiner Linux-Distro austauschen?

! Paketmanager sind so sehr mit dem System und den darauf laufenden Skripten verzahnt, dass ein Wechsel des Paketmanagers oder gar des Paketformats schwierig bis unmöglich ist. Einige Distributionen bieten die Möglichkeit, zusätzliche Paketmanager zu installieren. So liegt der Paketmanager Rpm auch in den Paketquellen von Ubuntu, dient dort jedoch nur zum Entpacken von RPM-Paketen.

Die parallele Nutzung von zwei Paketmanagern ist nicht ungefährlich, da sich die Programme ins Gehege kommen können. Klüger ist es, Pakete mit einem Paketkonverter wie Alien in das native Paketformat der eigenen Distribution umzuwandeln und dann zu installieren.

GPG-Schlüssel gesucht

? Wenn ich externe Paketquellen einbinde, bekomme ich immer wieder Fehlermeldungen und Hinweise auf fehlende GPG-Schlüssel. Was hat es damit auf sich?

! Zur Absicherung verwenden Paketmanager kryptografische Signaturen zur Verifikation der Pakete. Das soll sicherstellen, dass die Pakete nicht manipuliert wurden. Mit der gleichen Technik werden E-Mails signiert: Der Paketersteller signiert die Paketdatei mit seinem privaten Schlüssel und der Paketmanager überprüft diese Signatur dann anhand des öffentlichen Schlüssels des Erstellers.

Wenn dem Paketmanager kein öffentlicher Schlüssel bekannt ist, gibt er eine Fehlermeldung aus, da er die Integri-

tät des Paketes nicht verifizieren kann. Daher müssen Sie den Paketmanager anweisen, den Schlüssel zu importieren. Bei Apt passiert das mit dem Befehl `sudo apt-key adv --keyserver keyserver.ubuntu.com --recv SCHLÜSSEL-ID`, für Rpm lautet der Befehl `rpm --import SCHLÜSSEL-DATEI`. Achten Sie darauf, nur Schlüssel zu importieren, die vertrauenswürdig sind!

Apt und Apt-get

? Auf meinem Linux gibt es seit einiger Zeit sowohl den Befehl Apt als auch Apt-get; an manchen Stellen wird auch Aptitude erwähnt. Welchen soll ich nehmen?

! Apt wurde als vereinfachte Alternative zu Apt-get eingeführt. Apt macht die Paketverwaltung auf der Kommandozeile etwas simpler und spart außerdem ein wenig Getippe. Apt integriert einige Funktionen wie zum Beispiel die Suche nach Paketen, für die man sonst weitere Programme wie etwa Apt-cache braucht. Als Schmankerl bietet Apt außerdem eine grafische Fortschrittsanzeige. Es besteht aber erst mal kein Zwang zur Umstellung. Aptitude bietet neben den üblichen Funktionen eine grafische Schnittstelle, unterscheidet sich aus Anwendersicht jedoch nicht von Apt oder Apt-get.

Neue Pakete für alte Distros

? Ich setze bewusst eine ältere Distribution ein. Kann ich fehlende Software durch die Pakete einer neueren Version meiner Distro installieren?

! Mit Dpkg-basierten Distributionen ist das möglich, wenn auch nicht empfehlenswert. Sie müssen die Paketquellen der Nachfolgeversion zusätzlich zu den bestehenden in Ihr System einbinden und das Programm mittels `apt-get install PROGRAMM` installieren. Apt-get lädt dann die notwendigen Pakete herunter und aktualisiert bereits installierte Software bei Be-

Unter Debian kann man mit Checkinstall eine .deb-Datei aus einem Quellpaket erzeugen.

```

sakura
vim ~/.vim/vimrc x xterm-256color x merlin@mls-pc: /etc/pound x merlin@mls-debian: ~/zsh-5.2 x

merlin@mls-debian:~/zsh-5.2$ sudo checkinstall
[sudo] password for merlin:

checkinstall 1.6.2, Copyright 2009 Felipe Eduardo Sanchez Diaz Duran
Diese Software wurde unter der GNU GPL veröffentlicht

*****
*** Debian package creation selected ***
*****

Das Paket wird entsprechend dieser Vorgaben erstellt:

0 - Maintainer: [ root@mls-debian ]
1 - Summary: [ zshell ]
2 - Name: [ zsh ]
3 - Version: [ 5.2 ]
4 - Release: [ 1 ]
5 - License: [ GPL ]
6 - Group: [ checkinstall ]
7 - Architecture: [ amd64 ]
8 - Source location: [ zsh-5.2 ]
9 - Alternate source location: [ ]
10 - Requires: [ ]
11 - Provides: [ zsh ]
12 - Conflicts: [ ]
13 - Replaces: [ ]

Geben Sie die betreffende Nummer ein, um die Vorgaben zu ändern: █
    
```

darf. Nach Abschluss der Installation deaktivieren Sie die Paketquellen der Nachfolgeversion wieder. Jetzt betreiben Sie Ihre Distribution jedoch in einer Art Mischbetrieb und müssen diesen Vorgang regelmäßig wiederholen, da Sie sonst keine Updates für die Software erhalten. Überlegen Sie daher, ob ein Update des ganzen Systems nicht einfacher ist.

Die manuelle Installation von Einzelpaketen dürfte an den nicht erfüllbaren Abhängigkeiten scheitern. Falls sie doch gelingt, kann es sein, dass das Programm läuft. Besonders gut stehen die Chancen bei Programmen in Skriptsprachen, die erst zur Laufzeit kompiliert werden. Binärprogramme sind darauf angewiesen, dass Bibliotheken in einer bestimmten Fassung im System vorliegen. Wenn sich diese Bibliotheken geändert haben, wird auch das Binärprogramm nicht mehr starten.

Pakete im Selbstbau

? Die von mir benötigte Software fehlt in den Paketquellen meiner Linux-Distribution. Kann ich auch selbst ein Paket bauen?

! Ja, das ist aber nicht immer trivial. Wenn Sie lediglich eine aktualisierte

Version der Software benötigen und der Anbieter Quellcode-Patches bereitstellt, können Sie die Quellen Ihres Distributionspakets herunterladen, diese auf die von Ihnen benötigte Version „hochpatchen“ und dann in ein Paket wandeln. Dabei können aber Probleme mit Abhängigkeiten von aktualisierten Bibliotheken auftreten. Üblicherweise liegen der Dokumentation Ihrer jeweiligen Distribution spezifische Hinweise zum Paketieren von Software bei.

Oft lohnt auch ein genauer Blick auf die Projekt-Homepage der Software. Viele Projekte stellen eigene Paketquellen für verschiedene Distributionen bereit. Sie sollten sich genau überlegen, ob Sie die Software wirklich benötigen, denn im Fall von Sicherheitslücken liegt es an Ihnen, regelmäßig aktuelle Versionen einzuspielen. Sie können Ihre Pakete auch an die Gemeinschaft weitergeben, um anderen das Selbstpacken zu ersparen. Auch hierbei ist Vorsicht geboten, da die Sicherheit vieler anderer Anwender dann von der Vertrauenswürdigkeit Ihrer Pakete abhängt. *(mls@ct.de)*

Literatur

[1] Thorsten Leemhuis, Universalpakete, Flatpak und Snap sollen App-Installationen unter Linux revolutionieren, c't 17/16, S. 154