Debian 12 und Repositories aus dritter Hand

Das früher dafür verwendete Tool apt-key ist seit Jahren als veraltet markiert und sollte mit Debian 12 entfernt werden. Dazu die Manpage von apt-key:

```
apt-key(8) will last be available in Debian 11 and Ubuntu 22.04.
```

apt-key ist weiterhin als veraltet markiert und bei einem apt update wird die gleichen Warnungen erzeugt wie bisher:

```
W: https://wire-app.wire.com/linux/debian/dists/stable/InRelease: Schlüssel ist im veralteten Schlüsselbund trusted.gpg gespeichert (/etc/apt/trusted.gpg), siehe den Abschnitt MISSBILLIGUNG in apt-key(8) für Details.
```

Verwirrspiel

Einen Ersatz gibt es bisher nicht, was bedeutet, dass Schlüssel für Repositories aus dritter Hand manuell eingetragen und bereits vorhandene Repos händisch umgestellt werden müssen. Die Sache wird aber noch verwirrender, weil von Debian offiziell an verschiedenen Stellen unterschiedliche Handlungsvorgaben gemacht werden, wie mit den Schlüsseln von Repositories aus dritter Hand umzugehen ist.

Wie in der Warnung oben zu sehen ist, empfiehlt apt, diese Schlüssel anstatt wie bisher in /etc/apt/trusted.gpg abzulegen, das nun in /etc/apt/trusted.gpg.d zu tun. Im Debian Wiki steht dazu allerdings, das Hinzufügen von OpenPGP-Schlüsseln zu /etc/apt/trusted.gpg oder /etc/apt/trusted.gpg.d sei gleichermaßen unsicher. Dort wird /usr/share/keyrings empfohlen.

signed-by

Der Grund dafür ist, dass beim Hinzufügen eines OpenPGP-Schlüssels zum Signieren eines Repositorys zu einem der beiden Verzeichnisse dem Schlüssel von apt bedingungslos auch bei allen anderen auf dem System konfigurierten Repositorys vertraut wird, die über keine signed-by-Option verfügen. Infolgedessen kann jedes inoffizielle APT-Repository, dessen Signierschlüssel zu /etc/apt/trusted.gpg oder /etc/apt/trusted.gpg.d hinzugefügt wurde, jedes Paket auf dem System ungefragt aktualisieren oder ersetzen.

Deshalb ergibt es Sinn, sich bereits jetzt mit der im Debian-Wiki erwähnten Methode signed-by vertraut zu machen. Laut Debian-Wiki sollte der Schlüssel über HTTPS an einen Ort heruntergeladen werden, der nur von Root beschreibbar ist, (/usr/share/keyrings). Der Schlüssel sollte einen kurzen Namen erhalten, der das Repository beschreibt, gefolgt von archive-keyring als Erweiterung. Wenn das Repository z. B. mein_repository heißt, sollte die Schlüsseldatei mein_repository-archive-keyring.gpg heißen.

ASCII-Armor entfernen

Die OpenPGP-Keys von Repositories aus dritter Hand sind in der Regel mit der Methode ASCII-Armor versehen. Diesen Schutz gilt es bereits während des Downloads des Schlüssels mittels gpg –dearmor zu entfernen. Nehmen wir als Beispiel den Messenger Signal, der auf seiner Webseite bereits die korrekte Vorgehensweise vorgibt:

```
wget -O- https://updates.signal.org/desktop/apt/keys.asc | gpg --dearmor |
sudo tee /usr/share/keyrings/signal-archive-keyring.gpg
```

Nachdem der Schlüssel am richtigen Ort liegt, gilt es, den Eintrag für die Quellenliste zu formulieren. Um im Beispiel bei Signal zu bleiben, legen wir zunächst einen Listeneintrag in /etc/apt/sources.list/d an:

```
sudo nano /etc/apt/sources.list.d/signal.list
```

Der Inhalt des Eintrags sollte im Fall von Signal so aussehen:

```
deb [signed-by=/usr/share/keyrings/signal-archive-keyring.gpg]
https://updates.signal.org/debian/ stable main
```

Apt-Mitentwickler Julian Andreas Klose hat 2021 einen weiteren Vorschlag als Merge-Request eingebracht, dem das Quellformat DEB822 zugrunde liegt. Das habe ich aber bisher nicht weiter verfolgt.

Unklare Zukunft

Das Einbinden von Repositories aus dritter Hand wird mit Debian 12 und seinen Derivaten nicht unbedingt einfacher oder transparenter, aber angeblich sicherer. Mit der oben beschriebenen Methode gehören dann wenigstens die lästigen Warnungen von apt update der Vergangenheit an. Wenn man sich die Zeile, die den Schlüssel per wget holt und die Definition für die Quellenliste archiviert, muss man im Bedarfsfall nur noch die Befehle anpassen. Ob es zu einem späteren Zeitpunkt einen Ersatz für apt-key geben wird, bleibt unklar.