# Supply Chain Angriff & Paketsignaturen unter Debian

AS

IT Security Workshop

10/10/25

# Motivation

# Paketmanagement mit `apt`

- automatisierte (De-)Installation, Verwaltung und Aktualisierung von Softwarepaketen
- zu den offiziellen Repositories der Distribution können weitere Quellen hinzugefügt werden
  - öffentlichen Schlüssel für `apt` als vertrauenswürdig hinterlegen

# Supply Chain Angriffe

- **Problem**: Pakete und Quellen können in der Zukunft kompromittiert werden
- **zwei Beispiele aus 2024...**
  - typosquat Angriff auf die Open-Source Node Package Manager (NPM) Bibliothek
  - XZ Utils Backdoor

# Advanced Packaging Tools (APT)

- `apt`: Systemweite Installation von Paketen sowie ihrer Abhängigkeiten
  1. Download des `deb`-Archivs
  2. Verifikation der Integrität und Authentizität
  3. Auflösung von Abhängigkeiten
  4. Entpackung unter Root mittels `dpkg`
- Überprüfung auf und Installation von Updates
- `dpkg` verwaltet Datenbank aller installierter Pakete

# deb-Packages

- Beispielarchiv:

```
hello-world_1.0.0-1_arm64.deb
+--- DEBIAN/
|    +--- control
|    +--- preinst
|    +--- postinst
|    +--- ...
+--- usr/
     +--- bin/
          +--- hello-world
```

- control-Datei:

```
Package: hello-world
Version: 1.0.0
Maintainer: MyRepo <myrepo@myrepo.com>
Depends: libc6
Architecture: arm64
Description: Program printing "Hello World!"
```

# `apt`-Repositories

- Paketverzeichnis mit Index
- `Packages`-Datei:
  - Metainformationen über verfügbare Pakete mit Prüfsummen
- `Release`-Datei:
  - Metainformationen über Distribution mit Prüfsummen über Verwaltungsdateien
  - optional signiert mit OpenPGP

# Trust-Modell

- Tracking von vertrauenswürdigen Urhebern mit zugehörigen PGP-Schlüsseln
- keine Überprüfung von Signaturen auf Paketebene
- stattdessen: Verifikation der signierten `InRelease`-Datei
  - diese enthält Prüfsummen über dem Paketindex
  - der Paketindex enthält Prüfsummen über den Paketen
- Optional kann `apt` für die zusätzliche Paketsignatur mit `debsig-verify` konfiguriert werden

# Angriffspunkte

# Malicious Mirror

- Werden Schlüssel global als vertrauenswürdig in `/etc/apt/trusted.gpg.d/` hinterlegt, können diese theoretisch beliebige Pakete authentifizieren
  - auch solche fremder Repositories
- Alternativ kann in `/etc/apt/sources.list.d/<Repo>.list` der Schlüssel für die Quelle gesetzt werden.
  - Keine Notwendigkeit für globales Vertrauen.
  - Keine Möglichkeit, Daten des Repositories mit anderen Schlüsseln zu authentifizieren.

# Malicious Update

- `apt` erkennt bei identischem Paketnamen und höherer Versionsnr. ein Paket als Update
  - kein Tracking des Paketursprungs
- Updates für bereits installierte Pakete werden ohne Warnungen installiert, auch wenn diese aus einer anderen Quelle stammen.
- `apt`-**Pinning**
  - kann unerwünschte Updates verhindern
  - Regeln definierbar in `/etc/apt/preferences.d/`
  - Pinning von Paketen nach Release, Origin oder Version
  - Prio größer 999: Paket wird auch bei Downgrade installiert.

# Konzeptionelle Probleme:

- Überschreiben von Systemdateien ohne Warnungen
  - `dpkg` erkennt lediglich, wenn Dateien anderer Packages überschrieben werden.
- Ausführung von Pre-/Postinstall-Skripte mit `root`-Rechten

# Ausblick

# Snap

- Installation in eigener, isolierter Umgebung
- Anwendungen laufen in einem Container mit limitiertem Zugriff auf Host-System
- Abhängigkeiten werden mit dem Paket gebündelt
  - keine systemweite Installation
- baut auf Linux Security Module AppArmor auf

# SELinux

- Linux Security Module für Mandatory Access Control (MAC)
- Security Label für jedes Objekt im System
- Erlaubt Definition von Policies für Dateirechte abweichend von der nutzerbestimmten Rechtevergabe
- Bei Zugriffsversuch wird Security-Kontext des Prozesses überprüft
- Aber: experimentell und unpraktikabel für Debian-basierte Systeme, die stark auf App Armor setzen

# Referenzen

- https://wiki.ubuntuusers.de/Apt-Pinning
- https://debian-handbook.info/browse/en-US/stable/sect.package-authentication.html
- https://manpages.debian.org/testing/apt/apt-secure.8.en.html
- https://blog.packagecloud.io/how-to-gpg-sign-and-verify-deb-packages-and-apt-repositories/
- https://www.debian.org/doc/debian-policy/ch-controlfields.html
- https://en.wikipedia.org/wiki/Supply_chain_attack
- https://en.wikipedia.org/wiki/XZ_Utils_backdoor
- https://debian-handbook.info/browse/en-US/stable/sect.selinux.html