

Package management in Linux

TIJD VOOR IETS ANDERS?

De package managers onder Linux kennen zowel voor- als tegenstanders. Ze bieden snel toegang tot extra software of updates... totdat je een splinternieuwe of minder populaire applicatie zoekt! Met snaps zouden al die problemen opgelost zijn, maar is dat wel écht zo? > **Filip Vervloesem**

> Kenmerkend voor Linux zijn de verschillende package managers (zoals apt of yum) en hun grafische frontends. Via die package manager installeer én update je alle mogelijke software. Je hoeft dus geen installers van verschillende websites te downloaden. In de Windows- en macOS-wereld zijn het de ontwikkelaars die installatiepakketten voorzien voor hun eigen software. Bij Linux werkt dat anders: de ontwikkelaars leveren enkel de broncode, terwijl de distributeurs de installatiepakketten aanmaken.

Elke distributie bevat een geheel van pakketten die grondig getest én op elkaar afgestemd zijn. Veel programma's hebben immers specifieke versies nodig van bibliotheken (zoals OpenSSL of GTK), omdat ze functionaliteit uit die versie gebruiken. Voor de ontwikkelaars is het bijna onmogelijk om zelf installatiepakketten aan te bieden. Ze moeten dan immers voor elke versie van elke distributie een ander pakket maken met de correcte bibliotheekversies. Wil je de nieuwste versie van je favoriete programma draaien? Dan moet je wachten tot jouw distributie dat pakket voorziet! Meestal is dat pas bij de volgende grote update: dat betekent dus minstens een half jaar wachten...

SNAPS

Dankzij snaps kun je afzonderlijke programma's sneller updaten. Een snap bevat immers alle benodigde bibliotheken om het programma correct te draaien. Eenzelfde snap werkt dan ook op verschillende distributies én verschillende versies van elke distributie. Je ziet meteen waarom ontwikkelaars grote voorstanders zijn van snaps. Ze hoeven slechts eenmaal een installatiepakket aan te maken om alle gebruikers tevreden te stellen. Door snaps wordt de

> **Hecht je belang aan security, beperk je dan tot software uit de officiële repositories van je distributie** <

taak van de distributeur in feite overbodig gemaakt. Ontwikkelaars leveren hun software rechtstreeks aan de gebruikers en dus hoef jij niet te wachten tot de volgende update van jouw distributie. Een tweede voordeel is dat snaps meestal met erg beperkte rechten opstarten. Ze kunnen dus weinig kwaads uitrichten op het systeem, mochten ze schadelijke code bevatten.

NADELEN

Anderzijds mag je ook de risico's van snaps niet uit het oog verliezen. Het meeleveren van alle bibliotheken bij elke applicatie brengt immers verschillende nadelen met zich mee. Om te beginnen is dat gewoon niet zo efficiënt: snaps hebben meer schijfruimte nodig dan applicaties die je via de package manager installeert. Ook starten ze trager op, omdat ze geen gedeelde bibliotheken gebruiken die reeds in het geheugen zijn geladen. Een groter probleem is echter dat het de verantwoordelijkheid voor beveiligingsupdates volledig in schoenen van de ontwikkelaars schuift. Laten we als voorbeeld de Gimp eens bekijken. Dat is een behoorlijk grote applicatie met een 70-tal dependancies op externe bibliotheken. Wordt er een beveiligingslek ontdekt in pakweg libpcrc3, dan voorziet jouw distributie een beveiligingsupdate voor dat pakket.

UPDATES?

De volgende keer dat je de Gimp start, gebruik je meteen de gepatchte versie van die bibliotheek. Er is geen update nodig voor het Gimp-pakket: de Gimp-ontwikkelaars hoeven er zelfs niets voor te doen. Voorzien de ontwikkelaars zelf een snap, dan moeten ze die updaten zodra er een nieuwere versie is van hun applicatie óf van één van de 70 dependancies. Ze zijn dus niet enkel verantwoordelijk voor de updates van hun eigen code, maar ook voor die van de bibliotheken. Denk je dat alle ontwikkelaars de tijd en zin hebben om zich daarmee bezig te houden? Die kans is erg klein! Kortom: hecht je belang aan security, beperk je dan tot software uit de officiële repositories van je distributie. <

ADMIN CORNER