Software Assurance Tips

A product of the Software Assurance Tips Team[1]

Kevin Keen

Monday 3rd May, 2021

1 Downloading Package Dependencies for Offline Installs in Debian-based Distributions

Updated Friday 6th August, 2021

Sometimes in Software Assurance there exists the need to keep a machine from ever again connecting to a network. Perhaps a customer doesn't wish even the most remote risk that their code could be visible on a network. Or perhaps the analyst is examining malware. Or perhaps the results of a vulnerability scan need to be protected at a level higher than the network that is available. Reasons abound for keeping a machine offline, but when we need to install new software on these machines, it can be a harrowing experience. Those that have tried to install new packages offline in Debian-based environments are familiar with the "dependency hell" that quickly ensues. It is easy to overlook how much work a simple command like `apt-get` does until one tries to reproduce that effort manually.

One solution is to find one of various repository websites that list the package's dependencies and one by one visit each page to manually download them. While this can work, it is time-consuming and error-prone. A better solution is to let a utility like `apt` do the heavy lifting of identifying and downloading the dependencies in an automated manner.

Recently we had a similar situation. Although we had tried to find solutions in the past, most required scripts or third-party software. This time, however, we were able to find the command in Listing 1 on Stack Overflow.[2]

```
apt-get download \\
        $(apt-cache depends \\
                --recurse --no-recommends --no-suggests \\
                --no-conflicts --no-breaks --no-replaces \\
                --no-enhances <your-package-here> \\
                | grep ”^\w” | sort -u)
```

Listing 1: Deep Dependency Downloading

Breaking down the command, the portion inside the `$()` is executed first which searches the apt cache for dependencies of the selected package. The --no…options are generally needed to prevent optional or breaking packages from being included. Note that in some cases, these may need to be tweaked. For example, if there were a need to upgrade a package that is already installed on the offline machine, one may wish to remove the `--no-replaces` flag. Next, this output is passed to `grep` to filter out some lines of output that are not needed. Then, the output gets piped to `sort` with the `-u` flag in order to remove duplicates. Finally, `apt-get download` is called on each resulting line.

> ℹ Be sure that the online machine is the same OS version as the offline machine. Otherwise, the packages downloaded might not be compatible with the offline machine. Potentially with dire consequences!

This can quickly and easily allow a package and its dependencies to be downloaded for transfer to the offline machine.

References

[1] Jon Hood, ed. SwATips. https://www.SwATips.com/.

[2] Yann Vo. "How to list/download the recursive dependencies of a debian package?" In: (2019). URL: https://stackoverflow.com/a/41428445.