

## Packages - Bug #1310

### updating only pacman breaks system [caused by openssl upgrade]

2017-05-03 07:11 AM - oaken-source

<b>Status:</b> not-a-bug	<b>% Done:</b> 0%
<b>Priority:</b> critical	
<b>Assignee:</b>	
<b>Category:</b>	
<b>Description</b> pacman does currently not depend on openssl directly, which means currently it is possible to do `pacman -Sy && pacman -S pacman` without pulling in an updated libcrypto.  this results in the following error after the update:  <pre>pacman: error while loading shared libraries: libcrypto.so.1.1: cannot open shared object file: No such file or directory</pre> I propose that we add "openssl>=1.1" to the depends array of libre/pacman, to prevent this from happening.	
<b>Related issues:</b> Related to Packages - Bug #1299: pacman-5.0.1-5.parabola1 upgrade is broken <span style="float: right;">not-a-bug 2017-04-30</span>	

#### History

##### #1 - 2017-05-03 03:34 PM - isacdaavid

To be fair pacman does not depend on openssl; curl and libarchive do. This could be the beginning of a long slippery slope to try to change all packages to list their whole `pactree -u \$package` tree.

Call me a douche, but this is an opportunity to teach users [not to perform partial upgrades](#)

##### #2 - 2017-05-03 08:28 PM - libreuser

Any solution? :/

##### #3 - 2017-05-03 08:42 PM - lukeshu

Partial upgrades are not supported.

A user with a broken system can repair it by manually downloading whichever package they need, and extracting it to / with tar (they should then clean up the `/.PKGINFO` and friends files).

Edit: specifically, a package can be installed without pacman by doing `cd / && sudo bsdtar xf /path/to/package-ver.pkg.tar.xz && sudo rm .??*`

##### #4 - 2017-05-03 08:53 PM - lukeshu

- Status changed from open to not-a-bug

##### #5 - 2017-05-04 05:37 AM - oaken-source

hang on, if it is true that pacman does not depend on openssl directly, why do people get link errors, if they update pacman exclusively?

Isn't it recommended practice to update a very old system by:

```
pacman -Sy
pacman -S pacman
pacman -Su
```

this is precisely the workflow that breaks.

additional information:

```
$ lddtree /usr/bin/pacman
/usr/bin/pacman (interpreter => /lib64/ld-linux-x86-64.so.2)
  libalpm.so.10 => /lib64/libalpm.so.10
    libgpgme.so.11 => /lib64/libgpgme.so.11
      libassuan.so.0 => /lib64/libassuan.so.0
        libgpg-error.so.0 => /lib64/libgpg-error.so.0
```

```

libcurl.so.4 => /lib64/libcurl.so.4
libssh2.so.1 => /lib64/libssh2.so.1
libpsl.so.5 => /lib64/libpsl.so.5
libicuuc.so.59 => /lib64/libicuuc.so.59
libicudata.so.59 => /lib64/libicudata.so.59
libdl.so.2 => /lib64/libdl.so.2
libstdc++.so.6 => /lib64/libstdc++.so.6
libm.so.6 => /lib64/libm.so.6
libgcc_s.so.1 => /lib64/libgcc_s.so.1
libssl.so.1.1 => /lib64/libssl.so.1.1
libgssapi_krb5.so.2 => /lib64/libgssapi_krb5.so.2
libkrb5support.so.0 => /lib64/libkrb5support.so.0
libkeyutils.so.1 => /lib64/libkeyutils.so.1
libresolv.so.2 => /lib64/libresolv.so.2
libkrb5.so.3 => /lib64/libkrb5.so.3
libk5crypto.so.3 => /lib64/libk5crypto.so.3
libcom_err.so.2 => /lib64/libcom_err.so.2
libz.so.1 => /lib64/libz.so.1
libpthread.so.0 => /lib64/libpthread.so.0
libcrypto.so.1.1 => /lib64/libcrypto.so.1.1
libarchive.so.13 => /lib64/libarchive.so.13
libacl.so.1 => /lib64/libacl.so.1
libattr.so.1 => /lib64/libattr.so.1
libexpat.so.1 => /lib64/libexpat.so.1
liblzma.so.5 => /lib64/liblzma.so.5
liblz4.so.1 => /lib64/liblz4.so.1
libbz2.so.1.0 => /lib64/libbz2.so.1.0
libc.so.6 => /lib64/libc.so.6

```

as you can see, pacman is linked to libalpm, which is linked to libcrypto directly. libalpm is **part of pacman**:

```

$ pacman -Qo /lib64/libalpm.so
/usr/lib/libalpm.so is owned by pacman 5.0.1-5.parabolal

```

Hence I vote to reopen this issue.

#### #6 - 2017-05-04 07:48 AM - lukeshu

There seems to be some confusion.

The bug is closed as invalid because: Partial upgrades are not supported.

Even so, sometimes we do do nice things and make it harder for partial upgrades to break your system.

If pacman depended on openssl, this might be one of those times. We would change openssl to openssl>=1.1.

Now, regardless of whether or not pacman depends on openssl, and regardless of whether we do the depends=("openssl>=1.1") thing, this bug (bug [#1310](#)) is still invalid. Because partial upgrades are not supported.

Now, as for whether or not pacman depends on openssl:

Pacman is a frontend for libALPM (Arch Linux Package Manager). libALPM uses libcurl. libcurl uses libcrypto. When a library "links" against another library, it is common for the library itself to not actually be linked, but to communicate this to future compile-time linkers via pkg-config. This is what libcurl does; it puts Libs.private: -lcrypto in /usr/lib/pkgconfig/libcurl.pc rather than having /lib/libcurl.so literally link against libcrypto.

That is; just because ldd says that libalpm is linked directly against libcrypto does not mean that libcrypto is a non-transitive dependency of libalpm.

However libalpm DOES in fact depend on libcrypto for MD5 and SHA implementations. So it is a bug in the pacman PKGBUILD that openssl isn't listed as a dependency. This was an easy omission because it was masked by the transitive dependency on openssl by curl; I'm pretty sure that if you added openssl to the list, then namcap would even spit out a warning that you shouldn't be declaring transitive dependencies. However; this bug (the is-openssl-a-dependency-of-pacman bug) is in the Arch pacman package, and it should be reported to Arch.

#### #7 - 2017-05-04 04:16 PM - isacdaavid

- Related to Bug #1299: pacman-5.0.1-5.parabolal upgrade is broken added