

Installing the PulseSecure VPN client on Linux

Summary:

PulseSecure's VPN client is available in multiple operating systems, including Linux. The installs for Linux, however, are a little more crude as PulseSecure seems to have much less of an understanding on how to properly automate the process across a multitude of Linux platforms. This article will clarify install issues for the most popular Linux platforms - Ubuntu, Debian and Mint.

Linux Distribution Subtypes:

Today, there are over 300 "flavours" of Linux on the market - which is both a blessing and curse of being an open source OS. These different versions break down into a few basic subtypes of Linux, with the two largest base-types being Red-Hat and Debian. From those two are spawned a number of sub-distributions, and then from those spawn others and so on. Red-Hat tends to be more business oriented, breaking down into only a few popular subtypes. Red Hat itself as well as CentOS are largely business focused, where as Fedora is more user-focused. Red Hat Linuxs used the RPM package management system, and thus a user would need to download the .rpm version of PulseSecure's client. Debian, by far the most popular Linux base type, is used as the source of an enormous amount of different Linux versions, with the most popular two being Ubuntu and Mint. This is where things get a little more complicated. Ubuntu is based on Debian, so it can use .deb packages which is part of the Apt package management and installation system. Mint is based on Ubuntu, but also has a fairly heavy amount of development and customization to it's design. It also used the Apt system for package management. Mint tends to be the most popular of the distributions in use today.

The PulseSecure Linux Installer:

PulseSecure's understanding of Linux package managers and distributions in general seems very limited. Their installer is a very basic one that merely creates a directory, copies the PulseSecure client and files to it, and creates a menu entry. It manually shows dependencies but doesn't resolve them like it should (normally .deb and .rpm packages will deal with dependencies for you). It does provide a script to install the dependencies that you can run separately, but it doesn't always work depending on the Linux distribution you're using.

Installing on Ubuntu Linux:

PulseSecure builds and tests their installer on the primary version of Ubuntu (though presumably it works equally well on the official Ubuntu variations such as Ubuntu MATE, Xbuntu, Kubuntu, etc.). There are two ways to install the .deb package - via the GUI or via CLI. While the GUI is easier and likely what most people default to, the GUI does **not** show the output from the installation, so a user would not be aware that they need to install dependencies. To install via Command Line, open a Terminal and type the following command:

```
sudo dpkg -i pulse-8.2R4.1.i386.deb
```

Keep in mind that the filename will likely be different as new versions are released all the time. This will output the following:

```
Selecting previously unselected package pulse.
(Reading database ... 173765 files and directories currently installed.)
Preparing to unpack pulse-8.2R4.1.i386.deb ...
Unpacking pulse (8.2) ...
Setting up pulse (8.2) ...
Please execute below commands to install missing dependent packages
manually
apt-get install lib32z1
apt-get install libc6-i386
apt-get install libwebkitgtk-1.0-0:i386
apt-get install libproxy1:i386
apt-get install libproxy1-plugin-gsettings:i386
apt-get install libproxy1-plugin-webkit:i386
apt-get install libdconf1:i386
apt-get install dconf-gsettings-backend:i386
```

OR

You can install the missing dependency packages by running the below script

```
/usr/local/pulse/PulseClient.sh install_dependency_packages
```

Please refer /usr/local/pulse/README for instructions to launch the Pulse Client

The several "apt-get install <name>" lines are the commands needed to install the various dependences. Or, you can attempt to run the script included if you are familiar with running scripts in Linux. To install them manually, use "sudo" in front of each one to assume root power, like this:

```
sudo apt-get install lib32z1
```

Depending on the version of Ubuntu and sub-distribution type, some of these may already be installed or obsolete, so don't worry if not all of them install. Once these are done, the Pulse secure client is ready to run.

Installing PulseSecure on 64 bit Mint Linux (and other Debian/Ubuntu based versions):

Although Mint Linux is based on Ubuntu, there are some subtle differences that cause the PulseSecure installer to fail. If you try to run the installer via command line, you see something like this:

```
someone@computer ~/Downloads/Pulse $ sudo dpkg -i pulse-8.2R4.1.i386.deb
(Reading database ... 259564 files and directories currently installed.)
Preparing to unpack pulse-8.2R4.1.i386.deb ...
Do you want to clean up the connection store? [Yy/Nn] Y
Unpacking pulse (8.2) over (8.2) ...
Setting up pulse (8.2) ...
/var/lib/dpkg/info/pulse.postinst: line 235: [: =: unary operator expected
```

The installer has succeeded in placing the files and menu entries where they need to be - but fails as it tries to show the required dependencies. To install those manually, you will need to add the following commands in a command line terminal either one at a time like this:

```
sudo apt-get install lib32z1
sudo apt-get install libc6-i386
sudo apt-get install libwebkitgtk-1.0-0:i386
sudo apt-get install libproxy1:i386
sudo apt-get install libproxy1-plugin-gsettings:i386
sudo apt-get install libproxy1-plugin-webkit:i386
sudo apt-get install libdconf1:i386
sudo apt-get install dconf-gsettings-backend:i386
```

Or all at once like this:

```
sudo apt-get install lib32z1 libc6-i386 libwebkitgtk-1.0-0:i386
libproxy1:i386 libproxy1-plugin-gsettings:i386 libproxy1-plugin-webkit:i386
libdconf1:i386 dconf-gsettings-backend:i386
```

Depending on the version of Ubuntu and sub-distribution type, some of these may already be installed or obsolete, so don't worry if not all of them install. Once these are done, the Pulse secure client is ready to run.

Installing PulseSecure on 32-bit Mint Linux (and other Ubuntu/Debian based distributions)

Though 32 bit versions of Linux are not common anymore, older PCs that are not capable of running 64 bit OSs (Core Duo, Pentium Ms, Pentium4s and prior) have no choice but to run 32 bit OS code. The installation for these systems is pretty much the same with the exception that the installation of the additional dependencies require different package names. You can do them one at a time like this:

```
sudo apt-get install libc6
sudo apt-get install libwebkitgtk-1.0-0
sudo apt-get install libproxy1v5
sudo apt-get install libproxy1-plugin-gsettings
sudo apt-get install libproxy1-plugin-webkit
sudo apt-get install libdconf1
sudo apt-get install dconf-gsettings-backend
```

Or all at once like this:

```
sudo apt-get install libc6 libwebkitgtk-1.0-0 libproxy1v5
libproxy1-plugin-gsettings libproxy1-plugin-webkit libdconf1
dconf-gsettings-backend
```

Not all of the above may be required, so linux will install only what it needs. This will put in the required packages to make it work on the 32 bit version of Linux.

Configuring the PulseSecure Client:

On the menu system, an entry called "Other" should appear if not already there, and it is under this where you will find the PulseSecure client. You must edit this menu entry to ensure it is launched with the GUI-version of the sudo command that gives it elevated privileges, which is "gksu", so in the Command line, simply add that to the beginning it, leaving a space between it and the command used to launch the program. This may be already done depending on the version of the client installed; if so, leave it as is.

When creating a new connection, you can put whatever you wish for "Name", and for the URL in linux, use <https://remote-gate.uwo.ca/linux> - the /linux on the end is necessary to tell it that it can skip some of the steps used to further check Windows-based hosts. Once that's set, you are ready to connect and use the VPN.