By default, OpenVPN runs as the root user. This page seeks to describe how to instead run as an unprivileged user, "openvpn", instead. This is more secure than the built-in directives(--user and --group) because the openvpn process is never started with root permissions. Additionally, reconnects(including those which push fresh routes and configuration changes) which normally break after privileges are dropped via --user are handled without issue.

Configuration

Init Script

The init script is modifed to invoke the *openvpn* command via *su* instead of calling it directly(as root). It is recommended to copy the sample init script to a new one(**/etc/rc.d/init.d/openvpn-su**)before making these changes. Otherwise, package updates will wipe them out.

First, we must tell the init script which user to run as; insert the following near the top of the init script:

```
OPENVPN_USER="openvpn"
```

Next, remove the following line:

```
            $openvpn --daemon --writepid $piddir/$bn.pid --config $c --cd $work $script_security
```

....and replace it with:

```
            if [ -z "$OPENVPN_USER" ]
            then
                $openvpn --daemon --writepid $piddir/$bn.pid --config $c --cd $work $script_security
            else
                su $OPENVPN_USER -s /bin/sh --command="$openvpn --daemon --writepid $piddir/$bn.pid -
            fi
```

Optional: If you would like, you could move the OPENVPN_USER variable definition into a sysconfig file, and source that instead of defining it directly. This is more in line with typical init script behavior, where a different user may be desirable. The usage of the *if* block in the init script is meant to accommodate the possibility of the variable being undefined(in which case, openvpn will be executed as root).

Wrapper for *ip*

Because openvpn will be running unprivileged, it can't execute the *ip* command directly. Create a wrapper script, **/usr/local/sbin/unpriv-ip** (remember to chmod this to 755):

```
#!/bin/sh
sudo /sbin/ip $*
```

Next, grant sudo access to the openvpn user so it can use the wrapper script. Use *visudo* to edit your sudoers list, and insert the first line where convenient(at the end works well). NOTE: If you have previously specified "Defaults requiretty" in your sudoers(a useful additional security measure), you will need the second line as well.

```
openvpn ALL=(ALL) NOPASSWD: /sbin/ip
Defaults:openvpn !requiretty
```

TUN/TAP Device

Because openvpn will be running as an unprivileged user, a static tun/tap device is needed. The init script already supports running a shell script before executing openvpn, so create one to handle this task(**/etc/openvpn/openvpn-startup**):

```
#!/bin/sh
openvpn --rmtun --dev tun0
openvpn --mktun --dev tun0 --dev-type tun --user openvpn --group openvpn
```

User

If you are using openvpn from a binary distribution(such as that provided by EPEL), there should already be an openvpn user created, but it will need to be modified slightly. If it does not exist, create it.

```
[root@hostname ~]# mkdir /var/lib/openvpn
[root@hostname ~]# chown openvpn:openvpn /var/lib/openvpn
[root@hostname ~]# usermod -d /var/lib/openvpn -s /sbin/nologin openvpn
```

Some other directories will need to be set up so that the openvpn user can write to them.

```
[root@hostname ~]# mkdir /var/log/openvpn
[root@hostname ~]# chown openvpn:openvpn /var/run/openvpn /var/log/openvpn /etc/openvpn -R
[root@hostname ~]# chmod u+w /var/run/openvpn /var/log/openvpn -R
```

Config Changes

Lastly, you need to modify your openvpn config files to take advantage of all of these changes. Add the following directives to your openvpn configuration file(**/etc/openvpn/openvpn.conf**):

```
log /var/log/openvpn/openvpn
iproute /usr/local/sbin/unpriv-ip
dev tun0
persist-tun
```

Usage

Now, give it a whirl!

```
[root@hostname ~]# service openvpn-su restart
Shutting down openvpn:                                     [  OK  ]
Starting openvpn: Sun Dec  4 03:42:19 2011 TUN/TAP device tun0 opened
Sun Dec  4 03:42:19 2011 Persist state set to: ON
                                                           [  OK  ]
[root@hostname ~]# ps -ef |grep openvpn
openvpn  25557     1  0 03:42 ?        00:00:00 /usr/sbin/openvpn --daemon --wri
root     25560 25499  0 03:42 pts/0    00:00:00 grep openvpn
[root@hostname ~]#
```

Troubleshooting

Init Script

The init script changes above only apply to the default OpenVPN init scripts, not those provided by Debian/Ubuntu? and derivatives. These do not have support for the .sh auto-execute which this technique relies upon. You can try copying the default init script from the source distribution into **/etc/rc.d/init.d/openvpn-su** and then patching as above, but the author has not tested this methodology. Information and suggestions are welcomed.

Logs

Since openvpn is no longer being executed as root, it is unable to write to the syslog. Thus you must use **/var/log/openvpn/** and the *--log* directive. If no files are being created inside this directory, check that the permissions on the directory are correct(it should be owned by the openvpn user, and have a mask of 0755 / drwxr-xr-x).

Sudo

Permissions

You should also look at permissions/ownership for your keydir and **/etc/openvpn/**. The openvpn user should be able to read these, but not write to them, and no user but openvpn should be able to read your keys.

SELinux

In case you have SELinux enabled (e.g. you're using RHEL), you will need to set up additional user policies to allow the scripts run at startup. Create the following files:

```
# /tmp/openvpn_unpriv_hack.te

module openvpn_unpriv_hack 1.0;
```

```
require {
        type openvpn_t;
        type sudo_exec_t;
        class file { read open execute getattr execute_no_trans };
        class process setrlimit;
        class capability sys_resource;
}

#============= openvpn_t ==============
allow openvpn_t sudo_exec_t:file { read open execute getattr execute_no_trans};
allow openvpn_t self:process setrlimit;
allow openvpn_t self:capability sys_resource;
```

then compile and install the security modules:

```
$ checkmodule -M -m -o /tmp/openvpn_unpriv_hack.mod /tmp/openvpn_unpriv_hack.te
$ semodule_package -o /tmp/openvpn_unpriv_hack.pp -m /tmp/openvpn_unpriv_hack.mod
$ semodule -i /tmp/openvpn_upriv_hack.pp
```

and check if they have loaded correctly:

```
$ semodule -l | grep openvpn
openvpn 1.9.1
openvpn_unpriv_hack    1.0
```

Last modified on 10/18/12 13:03:38