How to load custom Kernel (tun) module in CentOS and RHEL Linux

Author : admin

Just recently it was necessery to load up a **tun** kernel module on few **CentOS** Linux servers.

I'm using Debian on daily basis, and everybody that had even little of experience with Debian should already be aware about the existence of the handy:
/etc/modules file.
On Debian to enable a certain kernel module to load up on Linux boot, all necessery is to just place the kernel module name in */etc/modules*.
For example loading the **tun** tunneling kernel module I issue the command:

```
debian:~# echo tun >> /etc/modules
```

I wondered if CentOS, also supports **/etc/modules** as it was necessery now to add this *tun* module to load up on CentOS's boot.
After a bit of research I've figured out CentOS does not have support for adding modules names in **/etc/modules** , anyhow after consulting **CentOS** documentation on *http://www.centos.org/docs/5/html/Deployment_Guide-en-US/s1-kernel-modules-persistant.html* , I found **CentOS and RHEL use /etc/rc.modules instead of Debian's /etc/modules** to load up any custom kernel modules not loaded by default during system boot.

Therefore instructing the RHEL Linux to load up my desired **tun** module in kernel on next boot was as easy as executing:

```
[root@centos ~]# echo 'modprobe tun' >> /etc/rc.modules
[root@centos ~]# chmod +x /etc/rc.modules
```

Now on next boot CentOS will load up the **tun** module in kernel. Achiving the same module load up is also possible through **/etc/rc.local** , but it's not recommended way as **/etc/rc.local** would load up the kernel module after all of the rest init boot scripts complete and therefore will load up the module slightly later, at the final boot stage.