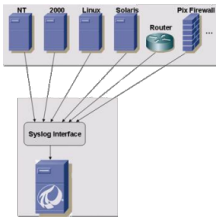


10 January 2007

By: Mihai Marinof, Linux Editor



## Setting Up A Central Syslog Server

*Make sure you have traceable evidence if anything goes wrong.*

One of the things I like most about Linux is that everything going on inside it is being logged. Whether a user logs in through ssh or a new visitor is passing through your website, everything is being logged in detail. However, the events related to the system are logged by a tool called **syslog**, which should be present on all Linux systems. Unfortunately, if a hacker breaks into your system, the first thing he'll probably try to do is cover all traceable tracks, rendering the logging tool useless. Basically, if the hacker is any good, you won't find any incriminating traces through the log files. You might even have a backdoor installed and not know anything about it. But this is where the syslog's *remote reception* feature comes in handy. You can set a syslog server on a safe, unused computer that will act as a central logging system for other computers (public computers that are more likely to be targeted) from the local network or even Internet. This way, all computers will immediately send any system-related events to the syslog server, ensuring that your logs will be completely accurate and un-tampered with at all times. This article will describe in detail how to set up a syslog server for one or more Unix systems, on Fedora Core and Ubuntu/Debian.

However, it *should* work for just about any Linux distribution. **Configure the syslog SERVER!** I'm sure most, if not all, Linux systems already have syslog installed so I'll skip this step. - First of all, you'll need to stop the syslog service: *Fedora Core:* `[CODE=0]service syslog stop[CODE=1]` *Ubuntu/Debian:* `[CODE=0]/etc/init.d/syslogd stop[CODE=1]` If you're running *another distribution* and these steps fail, also try: `[CODE=0]/etc/rc.d/rc.syslog stop[CODE=1]` and if it fails again, go for the old-school kill command: `[CODE=0]ps axfu | grep syslog[CODE=1]` copy the PID (number from second column) from the syslog line and: `[CODE=0]kill -9 PID[CODE=1]` Example: `[CODE=0]root:core][~]# ps axfu | grep syslogroot 12699 0.0 0.1 3884 668 pts/0 S+ 10:16 0:00 _grep syslogroot 12688 0.0 0.1 1692 576 ? Ss 10:12 0:00 syslogd -rm 0[root:core][~]# kill -9 12688[CODE=1]` - Next, you'll have to either edit the syslog start-up script to start syslog daemon with the "-r" flag, or manually start it with that flag. "-r" will enable remote reception feature, which will allow incoming logs. *Fedora Core:* - Open `/etc/sysconfig/syslog` with your favorite text editor - Find the line: `[CODE=0]SYSLOGD_OPTIONS="-m 0"[CODE=1]` - Replace it with: `[CODE=0]SYSLOGD_OPTIONS="-rm 0"[CODE=1]` - Restart the syslog daemon: `[CODE=0]service syslog restart[CODE=1]` *Ubuntu/Debian:* - Open `/etc/init.d/syslogd` with your favorite text editor - Find the line: `[CODE=0]SYSLOGD="-u syslog"[CODE=1]` - Replace it with: `[CODE=0]SYSLOGD="-ru syslog"[CODE=1]` - Restart the syslog daemon: `[CODE=0]/etc/init.d/syslogd restart[CODE=1]` On BOTH distributions you should see a message similar to "syslog restarted (remote reception) when executing the command: `[CODE=0]tail /var/log/messages[CODE=1]` On *other distributions* you should either find the RC syslog file, edit it and add the "-r" flag to the syslog options or, if you've used the old-school kill command, simply start syslog manually: `[CODE=0]syslogd -r[CODE=1]` - In the final step, you'll have to make sure the firewall isn't blocking any incoming packets. Simply run this iptables command so any rule will be overridden: `[CODE=0]iptables -I INPUT -p udp -i eth0 -s 192.168.1.2 -d 192.168.1.1 --dport 514 -j ACCEPT[CODE=1]` This rule will ensure that the syslog server (192.168.1.1) will receive UDP packets (containing log events) from the CLIENT (192.168.1.2). You MUST replace these IP addresses with the correct ones. Also, you will have to re-execute this command for every other client PC you may have (192.168.1.3, 192.168.1.4 etc). Then add this line (or lines) to rc.local so it will be executed every time the system boots. **Configure the CLIENT computers-** The client computers are configured to send any logged event to the syslog server, immediately as the events occur.

To do this, edit the file **/etc/syslog.conf** on every client computer and add this line AT THE TOP of the file:`[CODE=0]*.* @192.168.1.1[CODE=1]` Again, replace the example IP address with the syslog server's correct IP address. - Next, restart the syslog on every client you've edited:`[CODE=0]service syslog restart# or/etc/init.d/syslogd# or/etc/rc.d/rc.syslog restart# orps axfu | grep syslogkill -9 PIDsyslogd[CODE=1]` - Finally, make sure the client machine is allowed by the firewall to send UDP packets. Again, you can easily override any rule by running the iptables command:`[CODE=0]iptables -I OUTPUT -p udp -i eth0 -s 192.168.1.2 -d 192.168.1.1 --dport 514 -j ACCEPT[CODE=1]` Also, add this line to rc.local so it will be executed on every system boot. This is it. If everything was done correctly, you should start receiving log events to the syslog server. To view them, run:`[CODE=0]tail -f /var/log/messages(CTRL + C to escape)tail -f /var/log/secure(CTRL + C to escape)` `[CODE=1]` **KEEP IN MIND** that the machine running the central syslog MUST be secured to the fullest extent. If possible, use a machine that doesn't do much on your network so it won't capture attacker's attention, otherwise the whole purpose will be defeated.