System Resource Limits

By default, Linux will not impose any resource limits on user processes. This means any user is free to fill up all of the working memory on the machine, or spawn processes in an endless loop, rendering the system unusable in seconds. The solution is to set up some of your own resource limits by editing the /etc/security/limits.conf file:

```
$ sudoedit /etc/security/limits.conf
```

The possible settings are all explained in the comment within the file, and there are no silver bullet values to recommend, though we do recommend that you set up at least the `nproc` limit and possibly also the as/data/memlock/rss settings.

TIP **A Real-Life Resource Limit Example**

Just to give you an idea of what these limits look like on production servers, here is the configuration from the general login server of the Harvard Computer Society at Harvard University:

```
*       -       as              2097152
*       -       data            131072
```

```
*       -       memlock     131072
*       -       rss         1013352
*       hard    nproc       128
```

This limits regular users to 128 processes, with a maximum address space of 2GB, maximum data size and locked-in-memory address space of 128MB, and maximum resident set size of 1GB.

If you need to set up disk quotas for your users, install the quota package, and take a look at its man page.

System Log Files

As a system administrator, the system log files are some of your best friends. If you watch them carefully, you'll often know in advance when something is wrong with the system, and you'll be able to resolve most problems before they escalate.

Unfortunately, your ability to pay close attention to the log files dwindles with every server you're tasked with administering, so administrators often use log-processing software that can be configured to alert them on certain events, or they write their own tools in languages such as Perl and Python.

Logs usually live in /var/log, and after your server runs for a while, you'll notice there are a lot of increasingly older versions of the log files in that directory, many of them compressed with `gzip` (ending with the .gz filename extension).

Here are some log files of note:

- /var/log/syslog: general system log
- /var/log/auth.log: system authentication logs
- /var/log/mail.log: system mail logs
- /var/log/messages: general log messages
- /var/log/dmesg: kernel ring buffer messages, usually since system bootup

Your Log Toolbox When it comes to reviewing logs, you should become familiar with a few tools of choice. The `tail` utility prints, by default, the last ten lines of a file, which makes it a neat tool to get an idea of what's been happening last in a given log file:

```
$ tail /var/log/syslog
```

With the `-f` parameter, tail launches into follow mode, which means it'll open the file and keep showing you changes on the screen as they're happening. If you want to impress your friends with your new system administrator prowess, you can now easily recreate the Hollywood hacker movie staple: text furiously blazing across the screen.

Also invaluable are `zgrep`, `zcat`, and `zless`, which operate like their analogues that don't begin with a *z*, but on `gzip`-compressed files. For instance, to get a list of lines in all your compressed logs that contain the word "warthog" regardless of case, you would issue the following command:

```
$ zgrep -i warthog /var/log/*.gz
```

Your toolbox for dealing with logs will grow with experience and based on your preferences, but to get an idea of what's already out there, do an `apt-cache` search for "log files."

A Sprinkling of Network Security

Network security administration is another feature provided largely by the OS, so it's no different on Ubuntu than on any other modern Linux distribution. That means we won't cover it here but will leave you with a pointer.

The `iptables` command is the front end to the very powerful Linux firewall tables. Unfortunately, dealing with `iptables` can be rather difficult, particularly if you're trying to set up complex firewall policies. To whet your appetite, here's `iptables` in action, dropping all packets coming from a notorious time-sink domain:

```
$ sudo iptables -A INPUT -s www.slashdot.org -j DROP
```

Tutorials, how-tos, and articles about `iptables` are available on the Internet in large numbers, and the system man pages provide detailed information about all the possible options. Spending some time to learn `iptables` is well worth it because it'll let you set up network security on any Linux machine and will make it pretty easy for you to learn other operating systems' firewall systems if need be.

If you want to just manage a basic firewall on Ubuntu Server, you don't necessarily even need to venture into iptables. Ubuntu provides an excellent front-end called `ufw` that makes it very easy to add new firewall rules. For more information on `ufw`, check out the man page for that tool, or if you want a more complete reference, look at the security section of *The Official Ubuntu Server Book*.