My 10 Linux and UNIX Command Line Mistakes

Anyone who has never made a mistake has never tried anything new. — Albert Einstein.

Here are a few mistakes that I made while working at UNIX prompt. Some mistakes caused me a good amount of downtime. Most of these mistakes are from my early days as a UNIX sysadmin. This page lists my top ten Linux or Unix command line mistakes.

hey say, “Failure is the key to success; each mistake teaches us something.” I hope you will learn something from my 10 Linux or Unix command line mistakes as well as the comments posted below by my readers.

userdel Command

The file /etc/deluser.conf was configured to remove the home directory (it was done by previous sys admin and it was my first day at work) and mail spool of the user to be removed. I just wanted to remove the user account and I end up deleting everything (note -r was activated via deluser.conf):

```
userdel foo
```

Rebooted Solaris Box

On Linux killall command kill processes by name (killall httpd). On Solaris it kill all active processes. As root I killed all process, this was our main Oracle db box:

```
killall process-name
```

Destroyed named.conf

I wanted to append a new zone to /var/named/chroot/etc/named.conf file., but end up running:

```
./mkzone example.com > /var/named/chroot/etc/named.conf
```

Destroyed Working Backups with Tar and Rsync (personal backups)

I had only one backup copy of my QT project and I just wanted to get a directory called functions. I end up deleting entire backup (note -c switch instead of -x):

```
cd /mnt/bacupusbharddisk
tar -zcvf project.tar.gz functions
```

I had no backup. Similarly I end up running rsync command and deleted all new files by overwriting files from backup set (now I have switched to rsnapshot)

```
rsync -av -delete /dest /src
```

Again, I had no backup.

Deleted Apache DocumentRoot

I had sym links for my web server docroot (/home/httpd/http was symlinked to /www). I forgot about symlink issue. To save disk space, I ran rm -rf on http directory. Luckily, I had full working backup set.

Accidentally Changed Hostname and Triggered False Alarm

Accidentally changed the current hostname (I wanted to see current hostname settings) for one of our cluster node. Within minutes I received an alert message on both mobile and email.

```
hostname foo.example.com
```

Public Network Interface Shutdown

I wanted to shutdown VPN interface eth0, but ended up shutting down eth1 while I was logged in via SSH:

```
ifconfig eth1 down
```

Firewall Lockdown

I made changes to sshd_config and changed the ssh port number from 22 to 1022, but failed to update firewall rules. After a quick kernel upgrade, I had rebooted the box. I had to call remote data center tech to reset firewall settings. (now I use firewall reset script to avoid lockdowns).

Typing UNIX Commands on Wrong Box

I wanted to shutdown my local Fedora desktop system, but I issued halt on remote server (I was logged into remote box via SSH):

```
halt
service httpd stop
```

Wrong CNAME DNS Entry

Created a wrong DNS CNAME entry in example.com zone file. The end result – a few visitors went to /dev/null:

```
echo 'foo 86400 IN CNAME lb0.example.com' >> example.com && rndc reload
```

Failed To Update Postfix RBL Configuration

In 2006 ORDB went out of operation. But, I failed to update my Postfix RBL settings. One day ORDB was re-activated and it was returning every IP address queried as being on its blacklist. The end result was a disaster.

Conclusion

All men make mistakes, but only wise men learn from their mistakes — *Winston Churchill*.
From all those mistakes I have learn that:

1. You must keep a good set of backups. Test your backups regularly too.
2. The clear choice for preserving all data of UNIX file systems is dump, which is only tool that guaranties recovery under all conditions. (see Torture-testing Backup and Archive Programs paper).
3. Never use rsync with single backup directory. Create a snapshots using rsync or rsnapshots.
4. Use CVS/git to store configuration files.
5. Wait and read command line twice before hitting the dam [Enter] key.
6. Use your well tested perl / shell scripts and open source configuration management software such as puppet, Ansible, Cfengine or Chef to configure all servers. This also applies to day today jobs such as creating the users and more.

Mistakes are the inevitable, so have you made any mistakes that have caused some sort of downtime? Please add them into the comments section below.