Magical SysAdmin Incantations for New Freedom Fighters

For SCaLE 12x @ LAX Hilton
Audience: Beginner
Topic: SysAdmin
Presenter: George Robinson
Room: Century AB
Day: Saturday, February 22, 2014
Time: 15:00 to 16:00 PST

Magical SysAdmin Incantations for New Freedom Fighters

Welcome to a beginners guide to the basic spells cast by a first level linux sysadmin. Collected in this presentation are useful, everyday commands and programs that will impress your peers and confound your proprietary enemies. This presentation not only gives handy daily tips for linux administration, but also give and overview of why they are useful and how they work. Ultimately, the purpose is to strengthen the beginner linux system administrators to innovate and cast their own spells.

Who's this guy?!

The World's Most Interesting SysAdmin

- Computer hobbyist since I was 9
- First unix was in 1987
- Went pro with the internet boom
- Very fortunate to work in some great places with great talent

Why this talk?

- This is a small way of giving back to the community who has been so good to all of us
- This is to help beginners. Yes, I know there are better or more interesting ways to do what is included in this talk. Sometimes, it's sloppy on purpose.
- There is time left at the end of the talk for question and answer to better ground the concepts

Who is this for?

- This is for the new *nix admin, but assumes some basic linux and sysadmin knowledge
- Presented from a CentOS point of view, but concepts work just as well in any *nix
- The focus is not on basics, but on little tricks that are helpful in a pinch and illustrate concepts

This is boring. Insert a silly but informative picture.

Old timers unzip and unpack

- Check the contents

  ```
  gunzip < archive.tar.gz | tar -tvf -
  ```

- Unpack it: `gunzip < archive.tar.gz | tar -xvf -`
- You could just do `tar tzvf archive.tar.gz`
- “I need to download a bundle and unpack it and I like to live dangerously”

  ```
  wget -qO – "http://getmystuff.com/nonsense.gz" | tar zxvf -
  ```

Other old timey tricks

- `echo <<EOF` type in your nonsense and close with ^d (that's control+d doncha know!)
- Start program and send to background

  ```
  nohup md5sum bigfile.tgz 2>&1 &
  ```

  - `jobs, fg, ^Z, bg, nice`
  - See screen, dtach and disown for a different approach
- When all else fails - `man`

Network Info

- Show all open tcp ports and what pid they are directing to = `netstat -plant`
  - Does not account for xinet.d services
- Did they put me on the right vlan?
  - `tcpdump -vvv -nn -i eth0` and get ready to drink from the firehose
- Do I even have link? `ethtool eth0`

Network Info, p2

- Dang it, ethtool rejected me!

```
ifconfig eth0 192.51.100.0 netmask 255.255.255.0
```

- The network is flakey
  - Mtu size?
  - Bonded? Check via `cat /proc/net/bonding/bond0`
  - Force bond to an interface

```
/sbin/ifenslave -c bond0 eth1
```

`tcpdump` – intermediate magic

- -XX = ascii & hex
- -tttt = time stamp
- -vvv = very, very verbose
- -nn = numbers for IP and ports
- -l = line buffer
- -s = size
- -A = all packets
- -c = count

- How to drink from the firehose (YMMV):

```
tcpdump -tttt -vvv -nn -XX -l -s0 -c5
-A tcp src 8.8.8.8 and dst port 80 |
grep something
```

`lsof` - advanced magic

- LiSt Open Files and in *nix, everything is a file

  `lsof | grep`

- Grep for directories, users, files, pids or (deleted)

  `lsof -p <pid>`

  `lsof -N <nfs share>`

  `lsof -D <directory>`

`strace` – wizard, level 1

- `strace` and save output

  ```
  strace -o ouput <command>
  ```

- `strace` a running process

  ```
  strace -p <pid>
  ```

- -f = follow forks
- -e <system call>
- -t = time stamp
- -r = relative timing
- -c = count

- What is that program anyway?
  - Use `file`, `lsof` and `strings` for clues

Public Service Announcements

- Please take the time to learn just a bit of C – enough to modify something and then compile it and run it
- Bonus points for writing something in assembly
- It wouldn't hurt to have a basic electronics understanding as well, including transistor logic gates
- Save early, save often. Save the backups, save the world.
- Copy and paste will save you.
- Scripting can be accurate, and documented and is a requirement for repetitive tasks
- Version everything you touch
- Document as you go, or you won't. Some times sloppy docs are better than none.
- “Temporary solution” is an oxymoron

And now, back to our program...

The rich man's `locate`

- `locate` is the way to find a files on a system, but sometimes you're looking for more info
- All files modified 10 to 5 days ago

```
find /var/log -type f -mtime -10 mtime +5
```

- Now, do something with those files

```
find /home/me -type f -name *jpg* -mtime -10
-mtime +5 -print0 | xargs -r0 -P$(nproc)
-n10 md5sum
```

`grep` notes

- -v = grab non-matching lines
- -c = count
- -i = ignore case
- -l = list files that match
- -L = list files that don't match
- -r = recurse directories
- -H = print file name
- -h = hide file name
- -n = line number
- -C1 = show 1 line before and after match
- `grep 'this\|that'`

```
sudo grep --color -linR dhcp /var/log/*
sudo grep --color -i dhcp /var/log/messag
```

`vi` and you

- Vim is your color coding friend
- `:set list`
- `:set nu`
- `:%/s/^bad.*end/good\tend/g`
- `:n`, `n`, `.`
- `:w`, `:w!`, `:wq!` is bad!, `:q!` is good, `ZZ`, `view`
- `:%s/^V^M//g`, `:%s/^V^M/\r/g`, and I think `:%s/\r/\r/g`, or `` `dos2unix` ``
- http://vim-adventures.com/

This will be a little AWKward

- `cat <filename> | awk '{print $1, $NF}'`
- `find $PWD -type f -exec ls -althF {} \; | grep "Jan 26" | grep -v STGDB | grep PRDDB | awk '{print $6, $7, $8, "\t", $5, "\t", $NF}'`
- `find $PWD -type f -exec ls -altF {} \; | grep "Jan 26" | grep -v STGDB | grep PRDDB | awk '{sum+=$5} END {print sum}'`
- `find $PWD -type f -exec ls -altF {} \; | awk '{sum+=$5} END {print sum}'`

sed and tr

- I don't use them much on the command line, but it's good to know they're out there
- Delete leading and trailing white space
  - `sed 's/^[ \t]*//;s/[ \t]*$//'`
- Yet another way to convert dos to unix
  - `tr -d \r <infile >outfile`

tail, watch, wc

- To follow a file, `tail -f` or `less +F`
- Most recently touched files: `ls -altF | head`
- `watch -d -n10 wc -l sudo /var/log/messages`
- `watch -d -n2 du -sk * /tmp`

Sort & uniq

- `du -sk * | sort -rn | head`
- `du -sh * | sort -rh | head`
- `grep something <filename> | sort | uniq | wc -`
- `grep something <filename> | sort -u | wc -l`
- `grep something <filename> | sort | uniq -c`

Always on time: crontab

- It's handy to have a template in crontab

```
# .----------------- minute (0 - 59) OR /5 = every five min
# |  .-------------- hour (0 - 23) OR 8-5 = during business hours
# |  |  .----------- day of month (1 - 31) OR 1,15 = run on 1st and 15th
# |  |  |  .-------- month (1 - 12) OR jan,feb,mar,apr,may,jun,jul,aug,sep,oct,nov,
# |  |  |  |  .---- day of week (0 - 6) (Sunday=0 or 7)
# |  |  |  |  |         OR sun,mon,tue,wed,thu,fri,sat OR 1-5=weekdays
# *  *  *  *  *  command to be executed 2&>1 file.log
```

The ssh you may have missed

- `eval `ssh-agent`; ssh-add`
- `~.`
- Scp is biggie importante
- SecureCRT/FX – if they're going to make you use windows, make 'em pay for it! iTerm2 on OS X and terminator on *nix
- setup your remote host to accept ssh on ports 80 and 443
- Heck, you could set it up on every port, but only open ssh to the right port knock

Shell grab bag

- mailx -s "subject" me@spam.com
- sar, sar -r, sar -d, sar -f
- ls `cat list.txt`
- !$, !!, $?, $1, $*, $NF
- export HISTCONTROL=ignoreboth then, <space>command password
- history -d <num>
- reset

Shell grab bag, p2

- `chown -R <user>:<group>`
- `ls -d /*`
- `ln -s <file> <link>`
- `sudo !!`
- `chmod -reference <ref> <target>`
- `kill -HUP`
- `clear` or `^l` (lower L)

Cool little commands

- `wc -l`
- `bc`
- `^wrong^right`
- `cd -, cd ~`
- `curl`
- `(cd /tmp && ls)`
- `env`
- `\command` (unaliases)

Nice extras to have

- `fping`
- `mtr`
- `nmon`
- `mtr`
- `pv`
- `pee`
- `nc` (netcat)
- `links, elinks or lynx`
- `ifstat`
- `iftop`
- `iperf`
- `tip or minicom`
- `meld`
- `tcping`
- `hping`

Mandatory Line Noise

- This is my bash shell prompt. There are many like it, but this one is mine.

```
2014-02-21 21:38:55 george2@localhost:~
0$
```

- I put this in .bashrc

```
PS1="\[\e[00;32m\]\D{%Y-%m-%d}\[\e[00;37m\] \t
\[\e[00;33m\]\u\[\e[00;34m\]@\[\e[00;33m\]\h\
[\e[00;34m\]:\[\e[00;37m\]\w \n\[\e[00;31m\]\
$?\\[\e[00;37m\]\\$\[\e[00;37m\] \[\e[0m\]"
```

The End!

Any questions?