The Return of Unix Command-Line Kung Fu

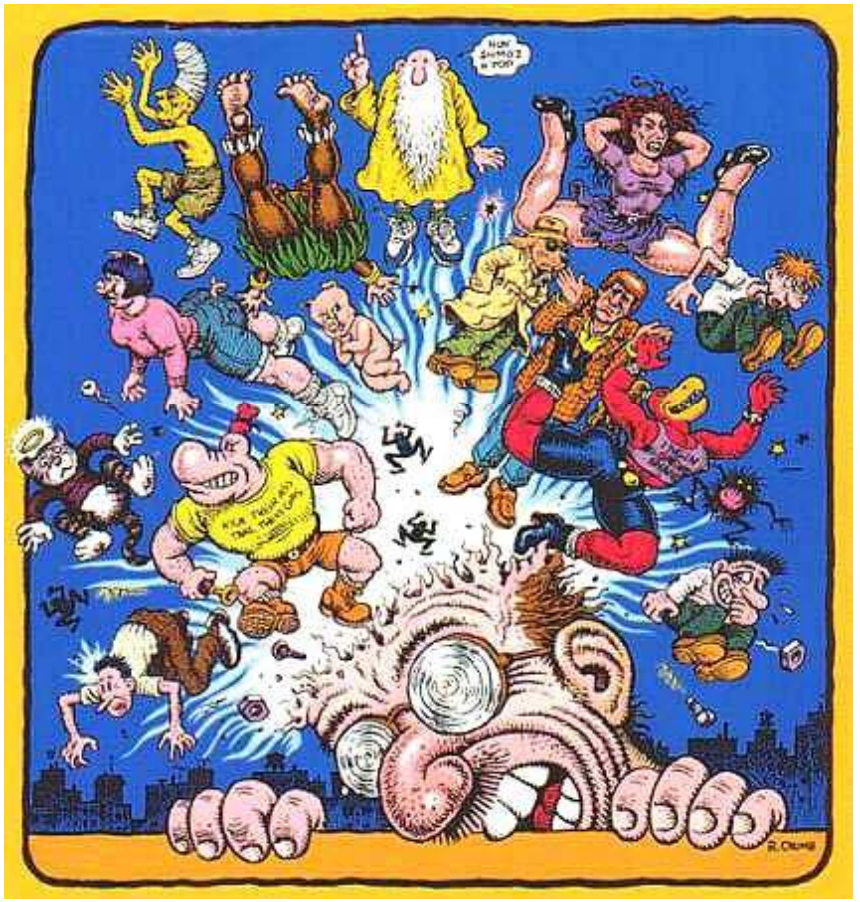
@Hal_Pomeranz

Who Is Hal Pomeranz?

- Independent consultant
- SANS Faculty Fellow, "oldest" surviving SANS instructor
  - Author, track lead for Sec506: Linux/Unix Security
  - Instructor for SANS Forensics classes
- Did I mention the blogs?
  - commandlinekungfu.com (w/ Ed Skoudis and Tim Medin)
  - http://blogs.sans.org/computer-forensics/

Let's Try Something Different

- Let's talk about problem-solving strategy
- Think of it as a look inside Hal's brain…
- Don't worry, there's plenty of neat shell tricks in there…



Problem #1: Human-Readable Time/Date

- Timestamps in epoch time? WTF?

```
type=USER_LOGIN msg=audit(1284648812.181:17): user
  pid=3344 uid=0 auid=500
  subj=system_u:system_r:unconfined_t:s0-s0:c0.c1023
  msg='uid=500: exe="/usr/sbin/sshd"
  (hostname=192.168.108.129, addr=192.168.108.129,
  terminal=/dev/pts/2 res=success)'
type=USER_AUTH msg=audit(1284648821.556:18): user
  pid=3389 uid=500 auid=500
  subj=user_u:system_r:unconfined_t:s0 msg='PAM:
  authentication acct="root" : exe="/bin/su"
  (hostname=?, addr=?, terminal=pts/2 res=success)'
```

- Would like to reformat each line with human-readable time and date stamp…

Step One: Break Problem Into Pieces

- Convert epoch times to human-readable format
- Extract epoch time value from line
- Combine the above two steps
- Loop through the entire file and format lines

Step Two: Work the Pieces

- Convert epoch times to human-readable format?

```
# date -d @1284649262
Thu Sep 16 08:01:02 PDT 2010
```

- Extract epoch time value from line

```
# tail -1 audit.log |
          sed 's/.*audit(\([0-9]*\).*/\1/'
1284649262
```

- Combine the above two steps

```
# date -d \
   @`tail -1 audit.log | sed 's/.*audit(\([0-9]*\).*/\1/'`
Thu Sep 16 08:01:02 PDT 2010
```

Step Two (cont): The Final Loop

```
# while read line; do
    echo $(date -d @$(echo $line |
                      sed 's/.*audit(\([0-9]*\).*/\1/')) \
         $line
  done <audit.log
Mon Jan 14 07:54:58 PST 2008 type=DAEMON_START
  msg=audit(1200326098.335:4111): auditd start, ver=1.5.5,
  format=raw, auid=4294967295 pid=1669 res=success, auditd
  pid=1669
Mon Jan 14 07:54:58 PST 2008 type=CONFIG_CHANGE
  msg=audit(1200326098.445:4): audit_enabled=1 old=0 by
  auid=4294967295 subj=system_u:system_r:auditd_t:s0 res=1
Mon Jan 14 07:54:58 PST 2008 type=CONFIG_CHANGE
  msg=audit(1200326098.559:5): audit_backlog_limit=320
  old=64 by auid=4294967295
  subj=system_u:system_r:auditctl_t:s0 res=1
[…]
```

Let's Try One from Jeff Haemer

- Directories named <*version*>-<*date*>:

```
$ ls
1.2.00.00_devel-20100906  2.0.00.00_devel-20100906
1.2.00.00_devel-20100907  2.0.00.00_devel-20100907
1.2.00.00_devel-20100908  2.0.00.00_devel-20100908
1.2.00.00_devel-20100909  2.0.00.00_devel-20100909
1.2.00.00_devel-20100910  2.0.00.00_devel-20100910
```

- Your task is to remove all but the *two most recent* copies of each version group

So What Are The Pieces?

- Get the version numbers
- Filter out the two most recent directories
- Remove the directories

Work The Pieces!

- Get the version numbers

```
$ ls | cut -d- -f1 | uniq
1.2.00.00_devel
2.0.00.00_devel
```

- Filter out the two most recent directories

```
$ ls -rd 2.0.00.00_devel* | tail -n +3
2.0.00.00_devel-20100908
2.0.00.00_devel-20100907
2.0.00.00_devel-20100906
```

- Remove the directories

```
$ for i in `ls | cut -d- -f1 | uniq`; do
     ls -rd $i* | tail -n +3
  done | xargs rm -rf
```

Cautionary Tale: The `/dev/tcp` Port Scanner

- Command output to network:

```
df >/dev/tcp/foo.example.com/9999
```

- Or a simple port checker:

```
$ echo >/dev/tcp/127.0.0.1/22
$ echo >/dev/tcp/127.0.0.1/23
bash: connect: Connection refused
bash: /dev/tcp/127.0.0.1/23: Connection refused
```

Cleaning Up That Last Example

- Why doesn't this work?

```
$ echo >/dev/tcp/127.0.0.1/23 2>/dev/null \
    && echo live || echo dead
bash: connect: Connection refused
bash: /dev/tcp/127.0.0.1/23: Connection refused
dead
```

- Need to do output redirection in a sub-shell:

```
$ (echo >/dev/tcp/127.0.0.1/23) 2>/dev/null \
    && echo live|| echo dead
dead
```

Beware Local Optimizations

- Simple port scanner:

```
$ for ((i=1; $i < 1024; i++)); do
    (echo >/dev/tcp/127.0.0.1/$i) 2>/dev/null \
        && echo $i/tcp live;
done
22/tcp live
631/tcp live
```

- But this is faster:

```
$ for ((i=1; $i < 1024; i++)); do
    echo >/dev/tcp/127.0.0.1/$i \
        && echo $i/tcp live;
done 2>/dev/null
22/tcp live
631/tcp live
```

Stumper Question

- Searching a directory structure
- Want to find files containing a particular string
- Only want to look in ASCII text files

```
find /usr/lib -type f | \
   xargs file | egrep -i 'script|ascii|text' | \
   cut -d: -f1 | xargs grep -l mystring
```

Finishing Up

- Any final questions?
- Questions later to: hal@deer-run.com
- Thanks for participating!

http://commandlinekungfu.com

http://blogs.sans.org/computer-forensics/

http://www.deer-run.com/~hal/