

Bootcamp: Linux

Peter Green

<http://agathongroup.com/talks/bootcamp08/>



Stuff We'll Cover

- Linux basics: shell, filesystem, users, processes, networking and system
- Configuring LAMP
- Cluster Architecture
- Packet Mangling with iptables
- “When Problems Attack”
- “Stupid Sysadmin Tricks”

Stuff We Won't Cover

- Email
- Writing code (Perl, Bash, PHP, Python, Ruby)
- Optimizing/fixing SQL
- Compilers, libraries, dependencies and other adventures in source code
- Revision control systems
- Network architecture
- Hardware
- File-sharing via NFS
- Licensing issues
- Network and system intrusion detection
- X-Windows
- VNC/remote GUI access
- Linux distribution wars
- vim versus emacs
- Compiled versus interpreted
- mutt versus pine versus elm
- bash versus tcsh
- Printing
- Samba, NCP, and the heterogeneous network
- Configuring and building a kernel from source
- BOOTP and diskless clients
- Jabber
- Time management
- Regular expressions
- sed, awk and other invaluable command-line utilities

First, a quiz

First, a quiz

- `ls`
- `ls -al`
- `ls -Al`
- `find . -type f | xargs ls -ld`
- `find . -type f -print0 | xargs -Or ls -ld`

First, a quiz

- `chmod 2775 /path/to/directory`
- `chmod 000 /path/to/file`
- `find . -name "*.php" -print0 | \`
`xargs -0 ls -ld | grep "Aug 22" | \`
`awk '{print $9}' | xargs grep 'mail('`

Linux Basics

Linux Basics: The shell

- What is the shell?
- Example shells
 - ▶ Bourne SHell: bsh
 - ▶ Bourne-Again SHell: bash
 - ▶ C SHell: csh
 - ▶ Korn SHell: ksh
 - ▶ DOS: command.com
- We'll use bash

Linux Basics: The shell

Paths

- Paths specify a list of directories to search
- PATH: executables
- LD_LIBRARY_PATH: loadable libraries
- CDPATH: directories for 'cd'
- MAILPATH: mail stores

Linux Basics: The shell

Aliases

- Map one command to another
- Common built-in aliases:
 - ▶ `alias ll='ls -l --color=tty'`
 - ▶ `alias cp='cp -i'`
- Other examples:
 - ▶ `alias vi='vim'`
 - ▶ `alias lsd='ls -Al | grep ^d'`

Linux Basics: The shell Builtins

- NOT commands
- Most common
 - ▶ cd, echo, history, kill, printf, pwd, ulimit
- Common programming builtins
 - ▶ “.”/source, break, continue, exit, getopt, read, shift, trap

Linux Basics: The shell

File handles and pipes

- Represented numerically
- stdin (0), stdout (1) and stderr (2)
- Upper limit governed by ulimit
- Current high: 595 file handles

Linux Basics: The shell

File handles and pipes

- `du -s * 2>/dev/null`
- `find . -type f >>files.txt 2>>files.txt`
- `find . -type f >>files.txt 2>&1`
- `find . -type f 2>&1 >>files.txt`

Linux Basics: The filesystem

- Collection of files
- What is a file in *nix?
 - ▶ Regular file or directory
 - ▶ Block or character special file
 - ▶ Executing text reference
 - ▶ Library
 - ▶ Stream
 - ▶ Network file (Internet socket, NFS file or UNIX domain socket)

Linux Basics: The filesystem

Permissions

- Octal math and masks: a primer
- Letters (with ops) or numbers
- Read (r/4), write (w/2), execute (x/1)
- User (u/100), group (g/10), other (o/1)

Linux Basics: The filesystem

Permissions

```
drwxr-xr-x 2 root root 4096 Jul 30 23:09 /usr/local/bin
-rwxr-xr-x 1 root root 938 Jul 30 23:09 /bin/blacklist.pl
-rw-r--r-- 1 root root 2161 Jul 30 23:09 /etc/passwd
-r----- 1 root root 1503 Jul 30 23:09 /etc/shadow

# chmod o-x /usr/local/bin/blacklist.pl
-rwxr-xr-- 1 root root 938 Jul 30 23:09 /bin/blacklist.pl

# chmod -x /bin/blacklist.pl
-rw-r--r-- 1 root root 938 Jul 30 23:09 /bin/blacklist.pl

# chmod a+x /bin/blacklist.pl
-rwxr-xr-x 1 root root 938 Jul 30 23:09 /bin/blacklist.pl
```

Linux Basics: The filesystem

Permissions

```
# chmod 400 /bin/blacklist.pl  
-r----- 1 root root    938 Jul 30 23:09 /bin/blacklist.pl  
  
# chmod 000 /bin/blacklist.pl  
----- 1 root root    938 Jul 30 23:09 /bin/blacklist.pl  
  
# chmod 755 /bin/blacklist.pl  
-rwxr-xr-x 1 root root    938 Jul 30 23:09 /bin/blacklist.pl  
  
# chmod 2755 /bin/blacklist.pl  
-rwxr-sr-x 1 root root    938 Jul 30 23:09 /bin/blacklist.pl  
  
# chmod 4755 /bin/blacklist.pl  
-rwsr-xr-x 1 root root    938 Jul 30 23:09 /bin/blacklist.pl
```

Linux Basics: The filesystem

Symlinks

- Symlink == “symbolic link”
- Special file: reference to another file
- Independent of target file

```
# ls -ld setlock  
lrwxrwxrwx 1 root root 16 Nov 9 2007 setlock -> /command/setlock
```

Linux Basics: The filesystem

Finding large files

- `du(1)`
 - ▶ A note on `man` and `command(N)`
- `-s`: summarize
- `-c`: total
- `-x`: one filesystem
- `-h`: human-readable

Linux Basics: The filesystem

Finding large files

```
[root@blackout ~]# du -s *
8      ag.sh
8      anaconda-ks.cfg
142260 centos-4-x86_64-default.tar.gz
8      drop.lasso
20     install.log
8      install.log.syslog
312    ipset
8      iptables-new.sh
24     rpmforge-release-0.3.6-1.el5.rf.x86_64.rpm
4320   src
8      tcrules.sh
8      t.pl
```

Linux Basics: The filesystem

Finding large files

```
[root@blackout ~]# du -sc *
8      ag.sh
8      anaconda-ks.cfg
142260 centos-4-x86_64-default.tar.gz
8      drop.lasso
20     install.log
8      install.log.syslog
312    ipset
8      iptables-new.sh
24     rpmforge-release-0.3.6-1.el5.rf.x86_64.rpm
4320   src
8      tcrules.sh
8      t.pl
146992 total
```

Linux Basics: The filesystem

Finding files by name

- `locate(1)`
- How it works
- Permissions: a caveat
- `-i`: case-insensitive

Linux Basics: The filesystem

Finding files by name

```
[pcg@blackout ~]$ locate agathongroup
```

```
[pcg@blackout ~]$ sudo locate agathongroup
```

```
Password:
```

```
/root/agathongroup
```

```
[pcg@blackout ~]$ sudo locate -i agathongroup
```

```
/root/AGATHONGROUP
```

```
/root/agathongroup
```

Linux Basics: The filesystem

Finding files: swiss army knife

- `find(1)`
- `find [path...] [expression]`
- `[expression]: -name [pattern]`
- `[expression]: -user [username]`
- `[expression]: -size [size]`
- `[expression]: ... anything?`

Linux Basics: The filesystem

Inspecting files

- `head(1)` and `tail(1)`
- `--lines=N`
- `--follow[=name]` (tail only)
- A brief history of pagers: `more(1)`
- `less(1)`

Linux Basics: The users

User accounts

- Username
- Password
- UID (User ID) / GID (Group ID)
- GECOS
- Home
- Shell

Linux Basics: The users

Passwords

- /etc/passwd

```
[pcg@blackout ~]$ grep pcg /etc/passwd  
pcg:pweh2l41HJKRB:500:500:~/home/pcg:/bin/bash
```

```
[pcg@blackout ~]$ grep pcg /etc/passwd  
pcg:x:500:500:~/home/pcg:/bin/bash
```

- /etc/shadow

```
[pcg@blackout ~]$ grep pcg /etc/shadow  
grep: /etc/shadow: Permission denied
```

```
[pcg@blackout ~]$ sudo grep pcg /etc/shadow  
pcg:pweh2l41HJKRB:13827:0:99999:7:::
```

Linux Basics: The users

Groups

- Groups facilitate file sharing

```
[root@blackout home]# groupadd staff
[root@blackout home]# gpasswd -a pcg staff
Adding user pcg to group staff
[root@blackout home]# mkdir staff
[root@blackout home]# chown root.staff staff
[root@blackout home]# chmod 2775 staff
[root@blackout home]# ll -d staff
drwxrwsr-x 2 root staff 4096 Oct 19 19:08 staff

[pcg@blackout staff]$ touch file
[pcg@blackout staff]$ mkdir directory
[pcg@blackout staff]$ ll
total 12
drwxrwsr-x 2 pcg staff 4096 Oct 19 19:10 directory
-rw-rw-r-- 1 pcg staff   0 Oct 19 19:10 file
```

Linux Basics: The processes

See what's running

- `ps(1)`
- Typical command line: `ps auxww`
- `a` = show all with tty
- `u` = display user-oriented format
- `x` = show all even without a tty
- `w` = more info, wrap lines

Linux Basics: The users

See what's running

```
[pcg@blackout ~]$ ps uaxww | grep blacklist
root      12273  0.0  0.0  26860  1204 ?        S      Aug22   47:15 /
usr/bin/perl /usr/local/bin/blacklist_newloop.pl
pcg       14139  0.0  0.0  61120   712 pts/0    S+     19:12   0:00
grep blacklist
root      21339  0.0  0.0   3620    84 ?        S      Feb27   0:00
supervise blacklist_cuda
root      21890  0.0  0.0   3768   396 ?        S      Sep23   0:22
multilog t s10000000 /var/log/blacklist_cuda
```

```
[pcg@blackout ~]$ ps uaxww | grep blacklist
root      12273  0.0  0.0  26860  1204 ?        S      Aug22   47:15 /usr/bin/perl /usr/local/bin/blacklist_newloop.pl
pcg       14139  0.0  0.0  61120   712 pts/0    S+     19:12   0:00 grep blacklist
root      21339  0.0  0.0   3620    84 ?        S      Feb27   0:00 supervise blacklist_cuda
root      21890  0.0  0.0   3768   396 ?        S      Sep23   0:22 multilog t s10000000 /var/log/blacklist_cuda
root      21904  0.0  0.0  22700  1616 ?        S      Sep23   3:35 /usr/bin/perl /usr/local/bin/blacklist_newcuda.pl
root      25109  0.0  0.0   3624    76 ?        S      Feb27   0:00 supervise blacklist_rbl
root      29876  0.0  0.0   3768   368 ?        S      Aug11   3:08 multilog t s10000000 /var/log/blacklist_rbl
```

Linux Basics: The processes

See what's listening

- `netstat(8)`
- Typical command lines:
`netstat -an | grep LISTEN`
`netstat -an | grep :80`

Linux Basics: The users

See what's listening

```
[root@twinpeaks root]# netstat -an | grep ^tcp | grep LISTEN
tcp        0      0 0.0.0.0:993          0.0.0.0:*          LISTEN
tcp        0      0 0.0.0.0:2626         0.0.0.0:*          LISTEN
tcp        0      0 0.0.0.0:322         0.0.0.0:*          LISTEN
tcp        0      0 0.0.0.0:514         0.0.0.0:*          LISTEN
tcp        0      0 0.0.0.0:995         0.0.0.0:*          LISTEN
[...]
```

```
[root@twinpeaks root]# netstat -an | grep :80
tcp        0      0 0.0.0.0:80           0.0.0.0:*          LISTEN
```

Linux Basics: The processes

See it all

- `lsof(8)`
- Typical command lines:
`lsof -n -i tcp:80`
`lsof /var/log/messages`
- Lots, lots more

Linux Basics: The users

See it all

```
[root@twinpeaks root]# lsof -n -i tcp:80
```

COMMAND	PID	USER	FD	TYPE	DEVICE	SIZE	NODE	NAME
httpd	1989	root	18u	IPv4	599494382		TCP	*:http (LISTEN)
[...]								

```
[root@twinpeaks root]# lsof /var/log/messages
```

COMMAND	PID	USER	FD	TYPE	DEVICE	SIZE	NODE	NAME
syslogd	874	root	2w	REG	9,0	85913	3460665	/var/log/messages

Linux Basics: The processes

See it in real-time

- `top(1)`
- “The top program provides a dynamic real-time view of a running system.”
- Generally no command-line options, but for fun...
- `nice -n -10 top -d.09`

Linux Basics: The users

See it in real-time

```
top - 19:37:37 up 344 days, 18:09, 1 user, load average: 1.46, 1.19, 0.99
Tasks: 110 total, 1 running, 109 sleeping, 0 stopped, 0 zombie
Cpu(s): 11.6%us, 18.7%sy, 0.0%ni, 56.6%id, 7.5%wa, 0.7%hi, 5.0%si,
0.0%st
Mem: 2058796k total, 2036416k used, 22380k free, 137120k buffers
Swap: 2040232k total, 35316k used, 2004916k free, 1129276k cached
```

PID	USER	PR	NI	VIRT	RES	SHR	S	%CPU	%MEM	TIME+	COMMAND
14506	root	15	0	13980	4372	4328	S	2	0.2	1329:28	pmacctd
26347	dnscache	15	0	243m	239m	372	S	2	11.9	183:06.95	dnscache
444	root	10	-5	0	0	0	S	1	0.0	2102:49	md1_raid1
3661	root	15	0	10072	712	592	D	1	0.0	457:35.47	syslogd
448	root	10	-5	0	0	0	D	0	0.0	1042:16	kjournald
2207	root	16	0	57584	644	536	S	0	0.0	321:18.75	sshd
3693	root	15	0	3768	392	344	S	0	0.0	356:26.12	klogd
12273	root	16	0	26860	1204	944	S	0	0.1	47:15.82	blacklist_newlo
14554	root	15	0	15508	5248	3324	S	0	0.3	229:48.67	pmacctd
14627	dnslog	18	0	3764	364	320	S	0	0.0	286:39.84	multilog
27240	root	18	0	58932	2616	2044	S	0	0.1	0:00.01	sshd
1	root	15	0	10316	84	48	S	0	0.0	0:44.75	init

Linux Basics: The network

Are you there?

- `ping(8)`
- `-n`: don't resolve names and IPs
- `-c [count]`: send [count] packets

```
[pcg@blackout ~]$ ping -c 1 -n agathongroup.com
PING agathongroup.com (209.151.235.164) 56(84) bytes of data.
64 bytes from 209.151.235.164: icmp_seq=1 ttl=64 time=0.176 ms

--- agathongroup.com ping statistics ---
1 packets transmitted, 1 received, 0% packet loss, time 0ms
rtt min/avg/max/mdev = 0.176/0.176/0.176/0.000 ms
```

Linux Basics: The network

How do I get to you?

- `traceroute(8)`
- `-n`: don't resolve names and IPs

```
[pcg@blackout ~]$ traceroute -n ais.cx
traceroute to ais.cx (64.78.150.4), 30 hops max, 40 byte packets
 1  209.151.228.9  0.567 ms  0.474 ms  0.475 ms
 2  4.71.36.225   94.615 ms  94.671 ms  94.728 ms
 3  4.68.20.126   9.270 ms  9.267 ms  9.278 ms
 4  4.69.137.37   7.690 ms  7.883 ms  7.861 ms
 5  4.69.132.9    10.223 ms 10.179 ms 10.163 ms
 6  4.69.134.226  18.393 ms 16.821 ms 13.340 ms
 7  4.69.134.209  9.572 ms 20.040 ms 20.194 ms
 8  4.69.132.58   35.479 ms 34.384 ms 34.175 ms
 9  4.68.107.66   34.446 ms 35.142 ms 34.623 ms
10  4.71.40.2     35.128 ms 35.102 ms 35.794 ms
```

Linux Basics: The network

Monitoring network traffic

- `tcpdump(8)`
- `-n`: don't resolve names and IPs
- `-i [iface]`: listen on interface [iface]
- `-A`: print packets in ASCII
- `-s 0`: print whole packets (0=unlimited)
- Expression-based packet matching

Linux Basics: The network tcpdump expressions

- port 80
- dst port 80
- src host 64.78.150.4 and dst port 80
- src host 64.78.150.4 and not port 22
- src net 64.78.150.0/26 and src port 80
- Lots, lots more

Linux Basics: The network

Monitoring network traffic

- One more note on tcpdump: redirect output!
- `tcpdump [...] >& file.log`
- `tcpdump -w file.log (then...)`
- `tcpdump -r file.log | less`

Linux Basics: The network

Scan for open ports

- `nmap(1)`
- First things first: play nice!
- Types of scans:
 - `-sS`: TCP SYN scan
 - `-sU`: UDP scan
 - `-sN/sF/sX`: TCP NULL, FIN, Xmas
 - `-sP`: ping scan
 - `-O`: enable OS detection

Linux Basics: The network

Scan for open ports

- `nmap -sS -O 192.168.0.1`
- `nmap -sP -v 192.168.0.0/24`
- Output options: XML, machine-readable, “`s | nc -l -p 4444`”

Linux Basics: The system

See the console

- `dmesg(8)`
- Boot messages
- Console messages
- See also `/var/log/dmesg` (system-dependent)

Linux Basics: The system CPU usage

- `uptime(1)` (ghetto!)

```
[pcg@blackout ~]$ uptime  
20:16:48 up 344 days, 18:48, 1 user, load average: 1.20, 1.06, 1.01
```

- `cat /proc/loadavg` (clunky!)

```
[pcg@blackout ~]$ cat /proc/loadavg  
0.73 0.97 0.98 1/108 1949
```

Linux Basics: The system Memory usage

- `free(1)`

```
[pcg@blackout ~]$ free
              total        used         free       shared    buffers     cached
Mem:          2058796    2038732         20064           0        137200    1125292
-/+ buffers/cache:    776240    1282556
Swap:         2040232         35316    2004916
```

- A word about “-/+ buffers/cache”
- A word about swap

Linux Basics: The system

Disk usage

- `df(1)`
- `-h`: human-readable

```
[pcg@blackout ~]$ df
Filesystem      1K-blocks      Used Available Use% Mounted on
/dev/md1        74610088    7839124  62919788  12% /
/dev/md0         101018      16012    79790    17% /boot
tmpfs           1029396         0    1029396   0% /dev/shm
```

```
[pcg@blackout ~]$ df -h
Filesystem      Size  Used Avail Use% Mounted on
/dev/md1        72G   7.5G   61G   12% /
/dev/md0         99M   16M    78M   17% /boot
tmpfs           1006M    0 1006M   0% /dev/shm
```

Linux Basics: The system

See it in real-time

- `sar(1)`
- Part of `sysstat` package
- Data collector: `sa`
- Data display: `sar`
- Historical: `sar [opts] -s [time] -e [time]`
- Real-time: `sar [opts] [freq] [num]`

Linux Basics: The system

See it in real-time

- -b: I/O and transfer rate statistics
- -I: IRQ statistics
- -n: network statistics
- -P: processor statistics
- -q: run queue (and load average)
- -r: memory and swap

Linux Basics: final words

- Don't be afraid to test stuff...
...and break stuff.
- Read man pages, --help output
- Find a guru, find a Linux Users Group
- Attend conferences, rub elbows
- Don't panic!

Configuring LAMP

Configuring LAMP: httpd

- httpd is Apache, “the most popular web server ... since April 1996”
- Managing config files
- Addressing virtual hosts
- Abusing error handlers
- mod_rewrite: The Swiss Army Knife of URL Manipulation

Configuring LAMP: httpd

Managing config files

- Include `/etc/httpd/conf.d/*.conf`

```
-rw-r--r-- 1 root root 279 Jun 8 12:28 alive.conf
-rw-r--r-- 1 root root 311 Aug 22 11:20 awstats.conf
-rw-r--r-- 1 root root 321 Jun 8 13:01 cac.conf
-rw-r--r-- 1 root root 349 Jul 23 09:48 cac.include
-rw-r--r-- 1 root root 303 Jun 8 12:34 creation.conf
-rw-r--r-- 1 root root 392 Jun 26 16:28 README
-rw-r--r-- 1 root root 9677 Jun 26 16:28 ssl.conf
drwxr-xr-x 2 root root 4096 Aug 10 15:30 ssl.crt
-rw-r--r-- 1 root root 286 May 7 15:21 ssl.include
drwxr-xr-x 2 root root 4096 Aug 10 15:29 ssl.key
-rw-r--r-- 1 root root 258 May 1 16:31 TEMPLATE
```

Configuring LAMP: httpd

Managing config files

- alive.conf: the typical config file

```
<VirtualHost *:80>  
  ServerName aliveupdates.com  
  ServerAlias www.aliveupdates.com  
  ServerAdmin webmaster@aliveupdates.com  
  DocumentRoot /home/alive/public_html  
  TransferLog logs/vhosts/alive/access-log  
  ScriptAlias /cgi-bin/ /home/alive/public_html/cgi-bin/  
</VirtualHost>
```

Configuring LAMP: httpd

Managing config files

- cac.conf: dealing with SSL

```
<VirtualHost *:80>
  Include conf.d/cac.include
</VirtualHost>
<VirtualHost 10.0.3.30:443>
  Include conf.d/cac.include
  Include conf.d/ssl.include
  SSLCertificateFile      conf.d/ssl.crt/www.comealivecruises.com.crt
  SSLCertificateKeyFile  conf.d/ssl.key/www.comealivecruises.com.key
</VirtualHost>
```

Configuring LAMP: httpd

Managing config files

- cac.include: dealing with SSL (cont.)

```
ServerName www.comealivecruises.com
ServerAlias comealivecruises.com
ServerAdmin webmaster@comealivecruises.com
DocumentRoot /home/cac/public_html
TransferLog logs/vhosts/cac/access-log
ScriptAlias /cgi-bin/ /home/cac/public_html/cgi-bin/
```

Configuring LAMP: httpd

Managing config files

- ssl.include: dealing with SSL (cont.)

```
SSLEngine on
SSLCipherSuite \
ALL:!ADH:!EXPORT56:RC4+RSA:+HIGH:+MEDIUM:+LOW:+SSLv2:+EXP:+eNULL
<Files ~ "\.(cgi|shtml|phtml|php3?)$" >
    SSLOptions +StdEnvVars
</Files>
SetEnvIf User-Agent ".*MSIE.*" \
    nokeepalive ssl-unclean-shutdown \
    downgrade-1.0 force-response-1.0
```

Configuring LAMP: httpd

Managing config files

- Use templates

```
<VirtualHost *:80>
  ServerName #DOMAIN#
  ServerAlias www.#DOMAIN#
  ServerAdmin webmaster@#DOMAIN#
  DocumentRoot /home/#USER#/public_html
  TransferLog logs/vhosts/#USER#/access-log
  ScriptAlias /cgi-bin/ /home/#USER#/public_html/cgi-bin/
</VirtualHost>
```

```
shell> sed 's/#DOMAIN#/ais.cx/g; s/#USER#/ais/g' < Tmpl > ais.conf
```

Configuring LAMP: httpd

Addressing virtual hosts

- IP address or wildcard?
- Non-SSL: wildcard
`NameVirtualHost *:80`
`<VirtualHost *:80>`
- SSL: IP address
`<VirtualHost 10.0.3.30:443>`
- Don't mix IPs and wildcards!
- Don't hardcode IPs unless necessary!

Configuring LAMP: httpd

Abusing error handlers

- ErrorDocument code document
- Serve static content (images, CSS) from the filesystem
- Set a 404 ErrorDocument to serve other content “dynamically”
- But this is an ugly solution...

Configuring LAMP: httpd

Abusing error handlers

- Defeats stats
- Allows for very simplistic pages only
- Creates additional load
- Inelegant
- There has to be a better way...

Configuring LAMP: httpd mod_rewrite

- mod_rewrite: The Swiss Army Knife of URL Manipulation
- Rewrite URLs transparently
- Proxy requests to backend servers
- Lots, lots more

Configuring LAMP: httpd mod_rewrite

- RewriteCond: environment vars

```
GeoIPEnable On
GeoIPScanProxyHeaders On
GeoIPDBFile conf.d/GeoIP.dat
RewriteEngine On
RewriteLog logs/rewrite_log
RewriteLogLevel 9
RewriteCond %{REQUEST_URI} !^/neutered-site
RewriteCond %{ENV:GEOIP_COUNTRY_CODE} ^NZ$
RewriteRule . http://%{SERVER_NAME}/neutered-site [R,L]
```

Configuring LAMP: httpd mod_rewrite

- RewriteMap: lookup tables

```
RewriteEngine On
RewriteLog logs/rewrite_log
RewriteLogLevel 9
RewriteMap chkip txt:/usr/local/share/chkip.txt
RewriteCond ${chkip:%{REMOTE_ADDR}} =1
RewriteRule . - [F]
```

```
# cat /usr/local/share/chkip.txt
64.78.150.4      1
64.78.150.2      1
64.78.150.1      0
```

Configuring LAMP: httpd mod_rewrite

- RewriteMap: lookup programs

```
RewriteEngine On
RewriteLog logs/rewrite_log
RewriteLogLevel 9
RewriteMap chkip prg:/usr/local/bin/chkip.pl
RewriteCond ${chkip:%{REMOTE_ADDR}} =1
RewriteRule . - [F]
```

```
#!/usr/bin/perl
while (<>) {
    $ip = $_; chomp $ip;
    if ($ip =~ /^64\.78\.150\./) { print "1\n"; next; }
    print "0\n";
}
```

Configuring LAMP: MySQL

- Flexible, extensible, scalable, high-performance, open-source “Standard Query Language” (SQL) database server
- Useful my.cnf variables
- Replication
- Up and coming

Configuring LAMP: MySQL

Useful my.cnf variables

- `old_passwords = 1`
 - ▶ Useful only when upgrading from MySQL 3.x
- `skip-innodb`
- `skip-bdb`
- `max_connections = N`
- `log-bin`
 - ▶ Necessary for replication, but useful even without

Configuring LAMP: MySQL

Useful my.cnf variables

- port = N
- socket = [path]
- *_buffer_size, *_cache_size...
- Server tuning... a black art?

<http://mysqltuner.com/mysqltuner.pl>

Configuring LAMP: MySQL

Useful my.cnf variables

- Sample .cnf files

<http://dev.mysql.com/doc/refman/5.0/en/server-parameters.html>

<http://dev.mysql.com/doc/refman/5.0/en/server-system-variables.html>

<http://dev.mysql.com/doc/refman/5.0/en/server-status-variables.html>

Configuring LAMP: MySQL Replication

- Take data and copy it elsewhere
- Data backup/security
- Database backups
- Analytics
- Geographic diversity

Configuring LAMP: MySQL Replication

- The Good
 - ▶ Asynchronous
 - ▶ Fast
 - ▶ Simple to switch to a new master
- The Bad
 - ▶ Only one master (usually!)
 - ▶ Asynchronous

Configuring LAMP: MySQL

Up and coming

- Circular (multi-master) replication
- MySQL Cluster
- MySQL Proxy

Configuring LAMP: PHP

- PHP – “PHP Hypertext Processor”
- “Widely-used general-purpose scripting language that is especially suited for Web development”
- Security
- Performance
- Contextual configuration

Configuring LAMP: PHP

Security

- “It’s getting better, a little better all the time.” — Paul
- “(It can’t get no worse!)” — John
- Security “versus” usability
- Fundamental problems lead to poorly-designed “fixes”

Configuring LAMP: PHP Security

- `safe_mode = ?`
- `register_globals = off`
 - ▶ Both going away in PHP6 — YAY!
- `mail()` is obsolete
 - ▶ PHPMailer
 - ▶ PEAR's Mail class
- Session handling

Configuring LAMP: PHP Security

<http://php-security.org/>

<http://hardened-php.net/>

<http://phpsec.org/>

<http://phpsecurity.org/>

“8 chapters. 30 exploits.”

[http://www.google.com/search?](http://www.google.com/search?q=PHP+security)

[q=PHP+security](http://www.google.com/search?q=PHP+security)

Configuring LAMP: PHP Performance

- `memory_limit = N`
- `register_long_arrays = false`
- Extensions
- PEAR

Configuring LAMP: PHP

Contextual configuration

- `php_value/php_admin_value`
- `php_flag/php_admin_flag`
- No PHP constants (`E_ALL`)!

<http://us.php.net/manual/en/ini.php#ini.list>

Cluster Architecture

Cluster Architecture: Concepts and Options

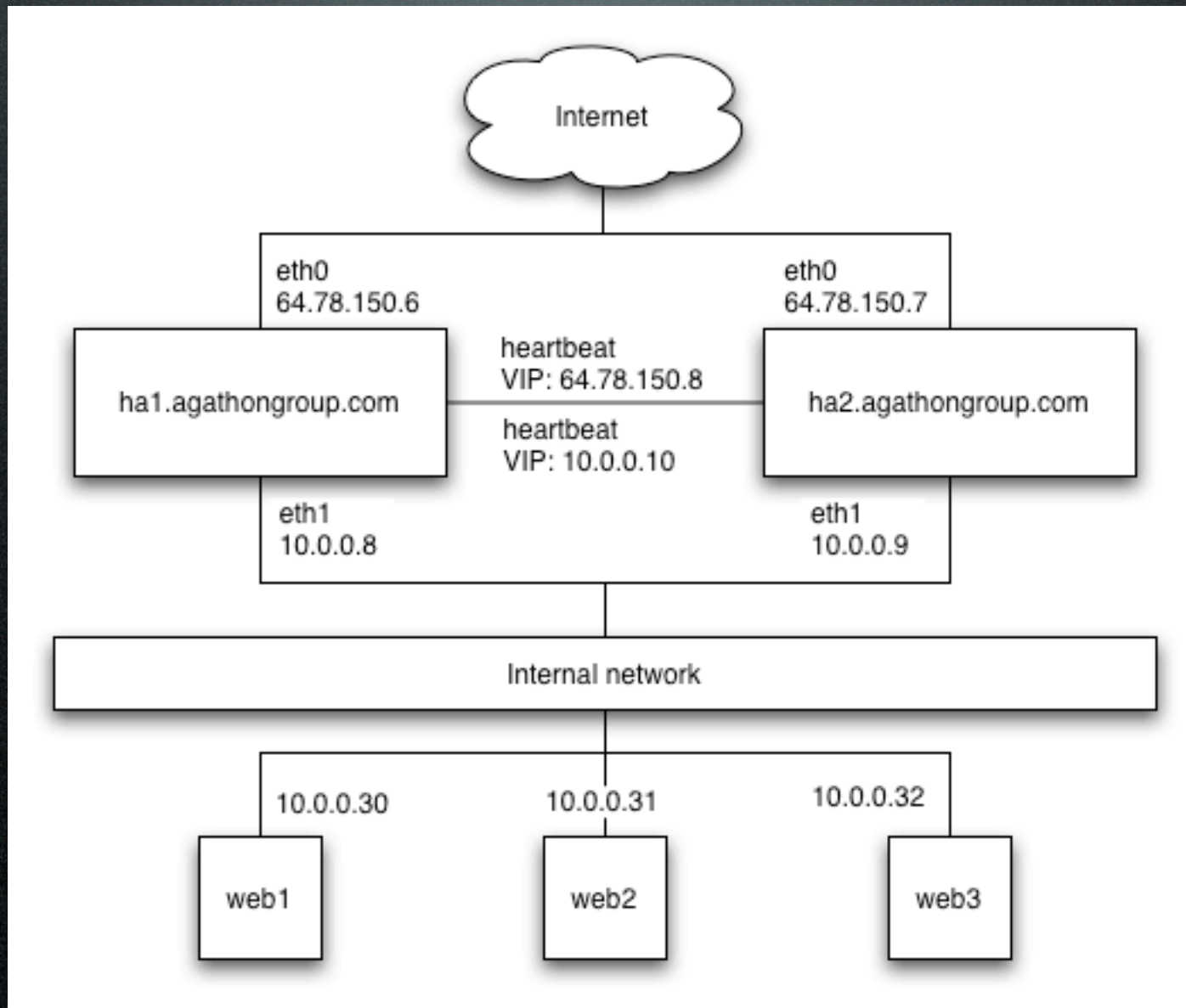
- High availability
- Load balancing
- Payware (e.g., F5 BIG-IP)
- Open Source (e.g., Ultramonkey)

Cluster Architecture:

Why cluster?

- Single server
 - ▶ Single point of failure
 - ▶ Simple (on-disk storage, programming)
- Cluster
 - ▶ Redundant systems
 - ▶ Complex (replicated/shared storage)

Cluster Architecture: Typical setup



Cluster Architecture: High availability (HA)

- heartbeats
- “Are you there?”
- One primary, one backup
- Simple: live web server, hot failover web server
- Modular: HA load balancers

Cluster Architecture: High availability (HA)

- Sample configuration (incomplete!)

```
ha.agathongroup.com \  
  IPaddr::10.0.0.10/8/eth1 \  
  IPaddr::64.78.150.8/26/eth0 ldirector::agathongroup.cf \  
  service::iptables \  
  service::ldap::restart
```

```
#baud 19200  
#serial /dev/ttyS0  
#bcast eth1  
#mcast eth0 225.0.0.1 694 1 0  
ucast eth1 10.0.0.8  
auto_failback on  
node ha1.agathongroup.com  
node ha2.agathongroup.com
```

Cluster Architecture: Load balancing

- Linux Virtual Service (LVS)
- 1+ servers behind a load balancer
- Identical content
- Potentially disparate servers

Cluster Architecture: Load balancing

- Simple web service

```
checktimeout=2
checkinterval=1
autoreload=no
logfile="/var/log/ldirectord/agathongroup.log"
quiescent=yes
virtual=64.78.150.8:80
    fallback=127.0.0.1:80
    real=10.0.0.30:80 masq
    real=10.0.0.31:80 masq
    checktype=connect
    scheduler=wrr
    protocol=tcp
```

Cluster Architecture: Load balancing

- Silly iptables+LVS trick: Use MARK to group packets for routing by LVS

```
$IPT -t mangle -A PREROUTING -p tcp -m multiport \  
--dports 25,110,143,993,995 -j MARK --set-mark 0x1
```

```
virtual=1  
real=10.0.0.4 masq  
service=smtp  
checktype=connect  
checkport=25  
scheduler=rr  
protocol=fwm
```

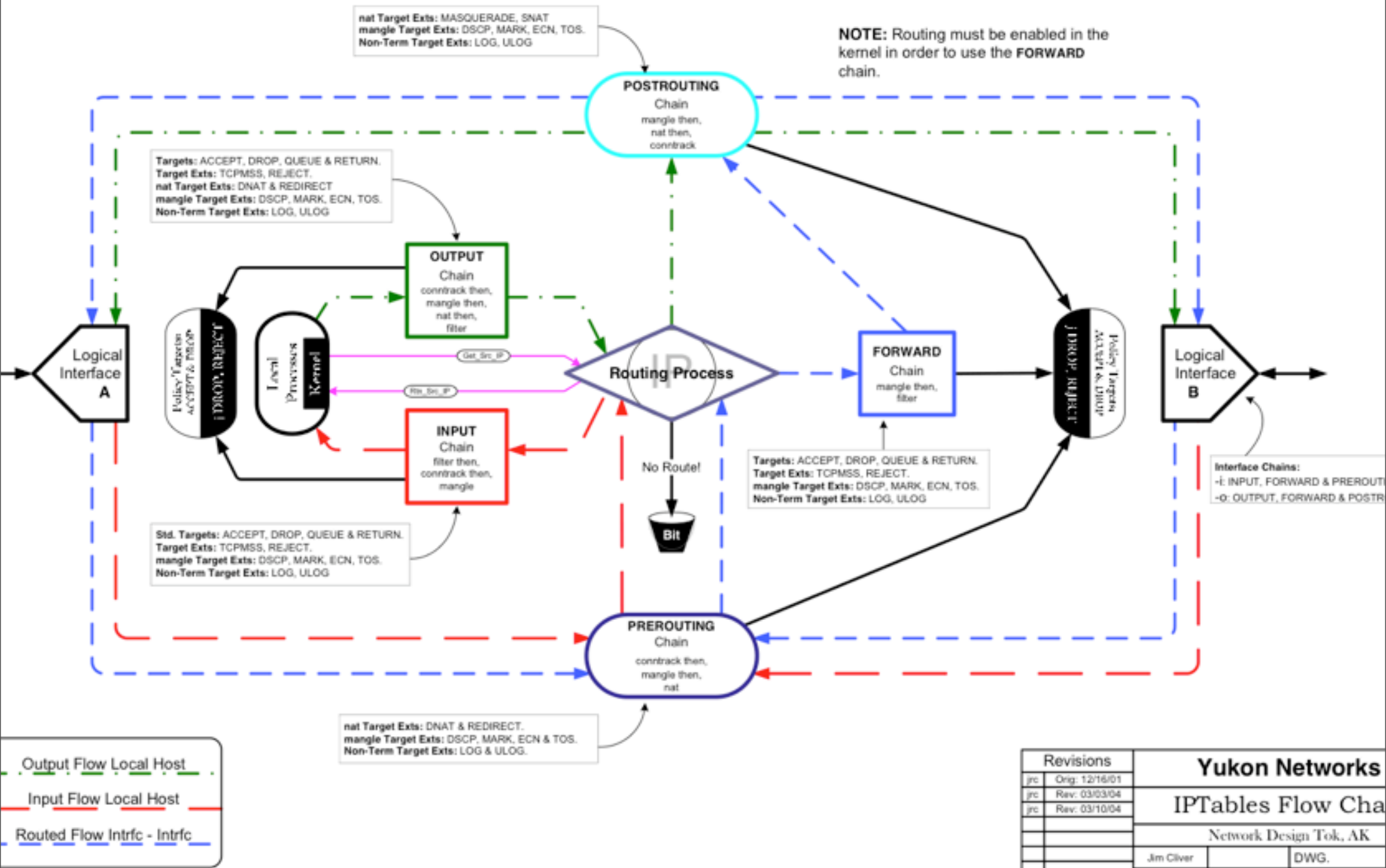
Cluster Architecture: Centralization

- Storage
 - ▶ Data synchronization
 - ▶ NFS
- Authentication, authorization and identification
 - ▶ Old school: NIS/YP
 - ▶ LDAP

Cluster Architecture: LDAP

- LDAP service
- Authentication and authorization:
pam_ldap
- Identification: Name Service
Switch (NSS) with nss_ldap

Packet Mangling with iptables



Revisions	
jc	Orig: 12/16/01
jc	Rev: 03/03/04
jc	Rev: 03/10/04

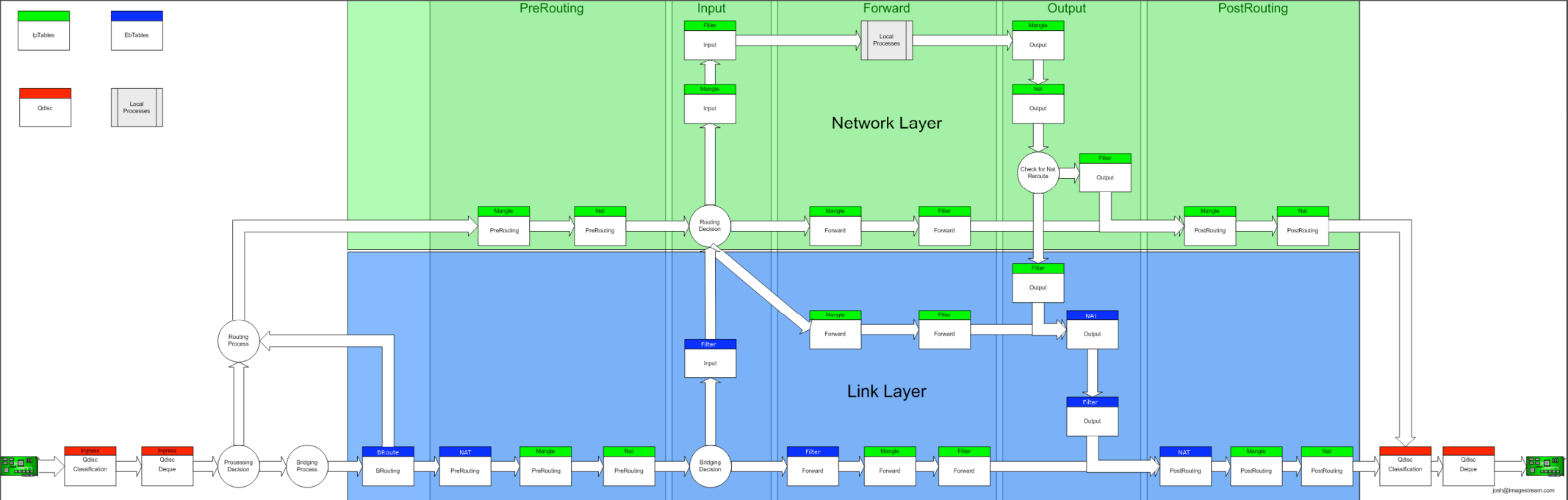
Yukon Networks

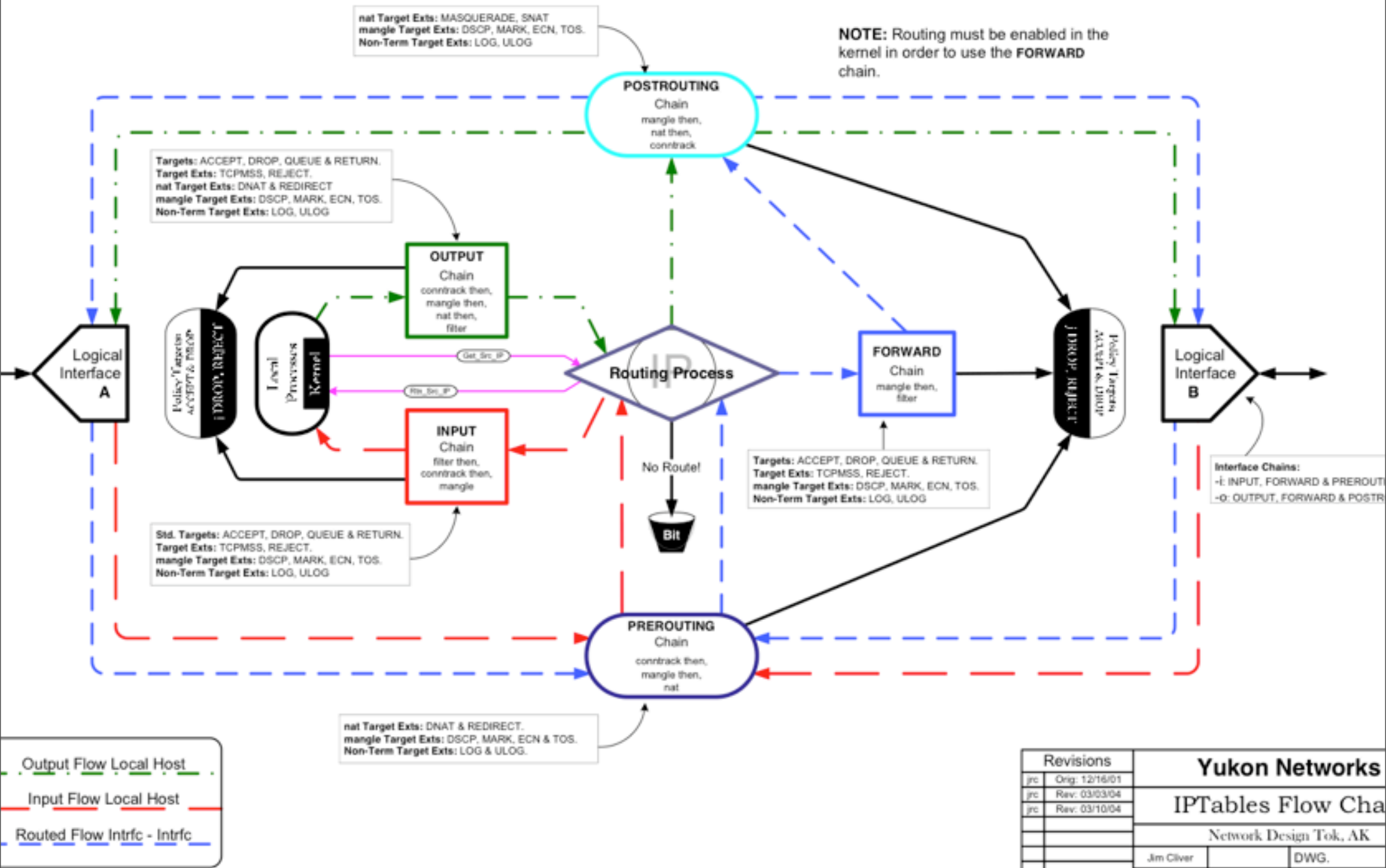
IPTables Flow Cha

Network Design Tok, AK

Jim Cliver DWG.

Scale: No Scale Sheet 1 of 1





Revisions	
jc	Orig: 12/16/01
jc	Rev: 03/03/04
jc	Rev: 03/10/04

Yukon Networks

IPTables Flow Cha

Network Design Tok, AK

Jim Cliver DWG.

Scale: No Scale Sheet 1 of 1

Packet Mangling with iptables: Packet flow

- mangle:PREROUTING
- nat:PREROUTING
- Routing decision
- (local) filter:INPUT
- (local) mangle:INPUT
- (non-local) mangle:FORWARD
- (non-local) filter:FORWARD
- (non-local) mangle:POSTROUTING
- (non-local) nat:POSTROUTING

Packet Mangling with iptables: Usage

- iptables [-AD] chain rule
- iptables -I chain [rulenum] rule
- iptables -D chain rulenum
- iptables -P chain target
- iptables -L chain

Packet Mangling with iptables: Common args

- [-sd] address[/mask]
- -p protocol
- [--sport/--dport] port[:port]
- -nv
- -j target
- -t table
- [-io] interface

Packet Mangling with iptables: “filter” table

- Used to filter traffic touched by the machine
- INPUT: packets destined for the local machine
- FORWARD: packets routing through the machine
- OUTPUT: packets generated on the local machine

Packet Mangling with iptables: “nat” table

- Used when a packet that creates a new connection is encountered
- PREROUTING: “as soon as packets come in”, before routing decision
- OUTPUT: locally-generated packets, before routing decision
- POSTROUTING: “as packets are about to go out”, after routing

Packet Mangling with iptables: “mangle” table

- Used for specialized packet alteration
- INPUT: packets destined for the local machine
- FORWARD: packets routing through the local machine
- OUTPUT: locally-generated packets, before routing decision

Packet Mangling with iptables: “mangle” table

- PREROUTING: “as soon as packets come in”, before routing decision
- POSTROUTING: “as packets are about to go out”, after routing

Packet Mangling with iptables: Built-in targets

- ACCEPT: let the packet through
- DROP: drop the packet
- QUEUE: pass the packet to a userspace program
- RETURN: stop in this chain and return the to “calling” chain

Packet Mangling with iptables: Extension targets

- DNAT: alter the destination address
`nat:PREROUTING` and `nat:OUTPUT` only
- (What about MASQUERADE?)
- SNAT: alter the source address
`nat:POSTROUTING` only
- LOG: log matching packets via the kernel log

Packet Mangling with iptables: Extension targets

- **FILTER**: like DROP, but send back an error packet in response
`filter:* only`
- **MARK**: set the netfilter mark
`mangle:* only`
- **REDIRECT**: like DNAT to the primary address of incoming interface
`nat:PREROUTING and nat:OUTPUT only`

Packet Mangling with iptables: Examples

- Simple router (DHCP)

```
IPT=/sbin/iptables
EXT_IF=eth0
INT_IF=eth1
INT_NET="192.168.1.0/24"
$IPT -t nat -A POSTROUTING -o $EXT_IF \
-s $INT_NET -j MASQUERADE
```

Packet Mangling with iptables: Examples

- Simple router (static, spoof protection)

```
IPT=/sbin/iptables
EXT_IF=eth0
INT_IF=eth1
INT_NET="192.168.1.0/24"
EXT_IP="64.78.150.1"
$IPT -A FORWARD -i $EXT_IF -s $INT_NET -j DROP
$IPT -A FORWARD -i $INT_IF '!' -s $INT_NET -j DROP
$IPT -t nat -A POSTROUTING -o $EXT_IF \
-s $INT_NET -j DNAT --to $EXT_IP
```

Packet Mangling with iptables: Examples

- Add “allow HTTP/S, deny all others”

```
IPT=/sbin/iptables
EXT_IF=eth0
INT_IF=eth1
INT_NET="192.168.1.0/24"
EXT_IP="64.78.150.1"
$IPT -A FORWARD -i $EXT_IF -s $INT_NET -j DROP
$IPT -A FORWARD -i $INT_IF '!' -s $INT_NET -j DROP
$IPT -A FORWARD -i $INT_IF -p tcp -m multiport \
--dports 80,443 -j ACCEPT
$IPT -A FORWARD -i $INT_IF -j DROP
$IPT -t nat -A POSTROUTING -o $EXT_IF \
-s $INT_NET -j DNAT --to $EXT_IP
```

Packet Mangling with iptables: Examples

- Log all outbound HTTP packets

```
IPT=/sbin/iptables
EXT_IF=eth0
INT_IF=eth1
$IPT -A FORWARD -i $INT_IF -o $EXT_IF \
-p tcp --dport 80 -j LOG -m limit --limit 20/min \
--log-prefix "FILTER "
```

Packet Mangling with iptables: Examples

- Drop XMAS/NULL packets

```
IPT=/sbin/iptables
```

```
EXT_IF=eth0
```

```
INT_IF=eth1
```

```
$IPT -A FORWARD -i $EXT_IF -o $INT_IF \
```

```
-p tcp --tcp-flags ALL ALL -j DROP
```

```
$IPT -A FORWARD -i $EXT_IF -o $INT_IF \
```

```
-p tcp --tcp-flags ALL NONE -j DROP
```

Packet Mangling with iptables: Examples

- “dn1” extension chain: creation

```
IPT=/sbin/iptables  
$IPT -N dn1  
$IPT -A dn1 -j LOG  
$IPT -A dn1 -j DROP
```

Packet Mangling with iptables: Examples

- “dnl” extension chain: better

```
IPT=/sbin/iptables
```

```
$IPT -N dnl
```

```
$IPT -A dnl -m limit --limit 20/min -j LOG
```

```
$IPT -A dnl -j DROP
```

Packet Mangling with iptables: Examples

- “dn1” extension chain: even better!

```
IPT=/sbin/iptables
```

```
$IPT -N dn1
```

```
$IPT -A dn1 -m limit --limit 20/min -j LOG
```

```
$IPT -A dn1 -j REJECT --reject-with tcp-reset
```

Packet Mangling with iptables: Examples

- before “dnf” extension chain

```
IPT=/sbin/iptables
```

```
EXT_IF=eth0
```

```
INT_IF=eth1
```

```
$IPT -A FORWARD -i $EXT_IF -o $INT_IF \
```

```
-p tcp --tcp-flags ALL ALL -j LOG
```

```
$IPT -A FORWARD -i $EXT_IF -o $INT_IF \
```

```
-p tcp --tcp-flags ALL ALL -j DROP
```

```
$IPT -A FORWARD -i $EXT_IF -o $INT_IF \
```

```
-p tcp --tcp-flags ALL NONE -j LOG
```

```
$IPT -A FORWARD -i $EXT_IF -o $INT_IF \
```

```
-p tcp --tcp-flags ALL NONE -j DROP
```

Packet Mangling with iptables: Examples

- after “dn1” extension chain

```
IPT=/sbin/iptables
EXT_IF=eth0
INT_IF=eth1
$IPT -A FORWARD -i $EXT_IF -o $INT_IF \
-p tcp --tcp-flags ALL ALL -j dn1
$IPT -A FORWARD -i $EXT_IF -o $INT_IF \
-p tcp --tcp-flags ALL NONE -j dn1
```

Packet Mangling with iptables: Examples

- blocking DoS attacks: the setup

```
IPT=/sbin/iptables
SPEC="-d 209.151.228.195 -p tcp -m tcp --dport 25"
$IPT -A FORWARD $SPEC -j allow_smtp
$IPT -A FORWARD $SPEC -j bl_smtp
$IPT -A FORWARD $SPEC --syn -j limit_smtp
$IPT -A FORWARD $SPEC --syn -j LOG \
--log-prefix "allow_smtp: "
```

Packet Mangling with iptables: Examples

- blocking DoS attacks: allow_smtp

```
IPT=/sbin/iptables
```

```
$IPT -A allow_smtp -s 64.78.150.0/26 -j ACCEPT
```

```
$IPT -A allow_smtp -s 209.151.228.192/26 -j ACCEPT
```

```
$IPT -A allow_smtp -s 209.151.235.128/26 -j ACCEPT
```

```
$IPT -A allow_smtp -s 204.253.132.0/23 -j ACCEPT
```

```
$IPT -A allow_smtp -s 64.78.177.0/24 -j ACCEPT
```

```
$IPT -A allow_smtp -s 64.233.160.0/19 -j ACCEPT
```

Packet Mangling with iptables: Examples

- blocking DoS attacks: bl_smtp

```
IPT=/sbin/iptables
```

```
$IPT -A bl_smtp -m recent --name bl_smtp \  
--rcheck -j dn1
```

```
shell> echo 1.2.3.4 > /proc/net/ipt_recent/bl_smtp
```

Packet Mangling with iptables: Examples

- blocking DoS attacks: limit_smtp

```
IPT=/sbin/iptables
$IPT -A limit_smtp \
-m hashlimit --hashlimit 10/min \
--hashlimit-burst 1 --hashlimit-mode srcip \
--hashlimit-name LIMITSMTP \
--hashlimit-table-expire 600000 -j RETURN
$IPT -A limit_smtp -j dn1
```

“When Problems
Attack”

“When Problems Attack”

Common complaints

- “My web site is down!”
- “My web site is slow!”
- “My web site is doing something funny!”

“When Problems Attack”

Common culprits

- Perception != reality
- What REALLY happened?
 - ▶ What did you do?
 - ▶ What did you expect to happen?
 - ▶ What actually happened?
 - ▶ What Do The Logs Say?TM

“When Problems Attack”

More common culprits

- Disk space
- Resource starvation
- Permissions
- DNS
- Caching

“When Problems Attack”

Where to look

- Console/dmesg
- System logs
- Application logs
- Diagnostic tools

“When Problems Attack”

Diagnostics: HTTP

```
# telnet mail.agathongroup.com 80
Trying 64.78.150.2...
Connected to mail.agathongroup.com (64.78.150.2).
Escape character is '^]'.
GET / HTTP/1.1
Host:mail.agathongroup.com

HTTP/1.1 302 Found
Date: Tue, 21 Oct 2008 04:05:18 GMT
Server: Apache/1.3.37 (Unix) mod_perl/1.29 PHP/4.4.4
Location: https://mail.agathongroup.com/
Transfer-Encoding: chunked
Content-Type: text/html; charset=iso-8859-1
X-Pad: avoid browser bug
```

“When Problems Attack”

Diagnostics: SMTP

```
# telnet mail.agathongroup.com 25
Trying 64.78.150.2...
Connected to mail.agathongroup.com (64.78.150.2).
Escape character is '^]'.
220 mail.agathongroup.com ESMTP
EHLO blackout.ais.cx
250-mail.agathongroup.com
250-AUTH LOGIN PLAIN
250-AUTH=LOGIN PLAIN
250-PIPELINING
250-STARTTLS
250 8BITMIME
MAIL FROM:<pcg@agathongroup.com>
250 ok
RCPT TO:<pcg@agathongroup.com>
250 ok
```

“When Problems Attack”

Diagnostics: SMTP

DATA

354 go ahead

From: pcg@agathongroup.com

To: pcg@agathongroup.com

Subject: test

test

.
250 ok 1224561588 qp 23844

QUIT

221 mail.agathongroup.com

Connection closed by foreign host.

“When Problems Attack”

Diagnostics: POP3

```
# telnet mail.agathongroup.com 110
Trying 64.78.150.2...
Connected to mail.agathongroup.com (64.78.150.2).
Escape character is '^]'.
+OK <26603.1224562089@twinpeaks.ais.cx>
USER pcg@agathongroup.com
+OK
PASS xxxxxx
+OK
QUIT
+OK
Connection closed by foreign host.
```

“When Problems Attack”

Diagnostics: IMAP

```
# telnet mail.agathongroup.com 143
Trying 64.78.150.2...
Connected to mail.agathongroup.com (64.78.150.2).
Escape character is '^]'.
* OK [CAPABILITY IMAP4rev1 UIDPLUS CHILDREN NAMESPACE
THREAD=ORDEREDSUBJECT THREAD=REFERENCES SORT QUOTA IDLE STARTTLS]
Courier-IMAP ready. Copyright 1998-2003 Double Precision, Inc. See
COPYING for distribution information.
001 LOGIN pcg@agathongroup.com xxxxxx
001 OK LOGIN Ok.
```

“When Problems Attack”

Diagnostics: POP3-SSL

```
# openssl s_client -connect mail.agathongroup.com:995  
CONNECTED(00000003)  
[lots of debugging output related to the SSL cert]  
+OK Hello there.
```

- IMAP-SSL, HTTPS: same
- STARTTLS (SMTP): -starttls smtp

“Stupid Sysadmin Tricks”

“Stupid Sysadmin Tricks”

Exploring /proc

- /proc: a real-time, memory-resident pseudo-filesystem that tracks the state of the system
- /proc/\$PID
- /proc/sys/net/ipv4

“Stupid Sysadmin Tricks”

Exploring /proc

- `/proc/scsi/*` and `/proc/ide/*`
- `/proc/cpuinfo`
- `/proc/mdstat`
- `/proc/net`
- `/proc/config.gz`

“Stupid Sysadmin Tricks”

Password-less ssh

- Goal: allow logins using keys, rather than passwords
- Local side: Build `~/.ssh/id_rsa.pub`
`ssh-keygen -N '' -t rsa`
- Remote side: copy `id_rsa.pub` into
`~/.ssh/authorized_keys`

“Stupid Sysadmin Tricks”

Password-less ssh

- RSA vs. DSA, SSHv1 vs. SSHv2
- sshd_config:
PubkeyAuthentication yes
- Permissions: go-wx on:
~/.ssh/authorized_keys/
~/.ssh/
~

“Stupid Sysadmin Tricks”

Booting utilities

- Knoppix live CD
- linux rescue
- `init=/bin/sh`

“Stupid Sysadmin Tricks”

Useless Use of cat

- `cat file | less`
`less file`
- `cat file | wc -l`
`wc -l file` OR `wc -l < file`
- `for i in `cat file`; do echo $i; done`
`while read i; do echo $i; done < file`
- `for i in `cat servers`; do \
ssh $i uptime; done`
`xargs -i ssh {} uptime < servers`

“Stupid Sysadmin Tricks”

UseFUL Use of cat

- `(foo ; bar ; cat file ; baz) | quux`
- `cat file1 file2 | wc -l`
- `cat<<EOF`
`data`
`EOF`
- `cat` == “concatenate”! In general, don’t use with `<2` files.

“Stupid Sysadmin Tricks”

Sorting

- `sort(1)`
- `-n`: numeric
- `-r`: reverse
- `-k [POS]`: sort on the POSth field
- `-t [SEP]`: use SEP as field separator

“Stupid Sysadmin Tricks”

Sorting

```
[root@blackout ~]# du -s *
8      ag.sh
8      anaconda-ks.cfg
142260 centos-4-x86_64-default.tar.gz
8      drop.lasso
20     install.log
8      install.log.syslog
312    ipset
8      iptables-new.sh
24     rpmforge-release-0.3.6-1.el5.rf.x86_64.rpm
4320   src
8      tcrules.sh
8      t.pl
```

“Stupid Sysadmin Tricks”

Sorting

```
[root@blackout ~]# du -s * | sort -nr
142260 centos-4-x86_64-default.tar.gz
4320   src
312    ipset
24     rpmforge-release-0.3.6-1.el5.rf.x86_64.rpm
20     install.log
8      t.pl
8      tcrules.sh
8      iptables-new.sh
8      install.log.syslog
8      drop.lasso
8      anaconda-ks.cfg
8      ag.sh
```

“Stupid Sysadmin Tricks”

Sorting

```
[root@blackout ~]# du -sh * | sort -nr
312K    ipset
139M    centos-4-x86_64-default.tar.gz
24K     rpmforge-release-0.3.6-1.el5.rf.x86_64.rpm
20K     install.log
8.0K    t.pl
8.0K    tcrules.sh
8.0K    iptables-new.sh
8.0K    install.log.syslog
8.0K    drop.lasso
8.0K    anaconda-ks.cfg
8.0K    ag.sh
4.3M    src
```

- Oops!

“Stupid Sysadmin Tricks”

Sorting

```
[root@blackout ~]# du -s * | sort -r -k2
8      t.pl
8      tcrules.sh
4320   src
24     rpmforge-release-0.3.6-1.el5.rf.x86_64.rpm
8      iptables-new.sh
312    ipset
8      install.log.syslog
20     install.log
8      drop.lasso
142260 centos-4-x86_64-default.tar.gz
8      anaconda-ks.cfg
8      ag.sh
```

“Stupid Sysadmin Tricks”

sudo

- Why sudo rules
 - Dangers of logging in as root
 - Audit trail
 - Granular permissions
- Incorrect uses of sudo
 - NOPASSWD
 - sudo su -

“Stupid Sysadmin Tricks”

sudo

- Where sudo falls short
 - `foo > /root/file`
 - `foo | sudo tee /root/file`
 - `foo < /etc/shadow`
 - `sudo cat /etc/shadow | foo`
 - UUOC!
 - Not really
- Fast typing can lead to mistakes similar to those of being logged in as root

“Stupid Sysadmin Tricks”

One-liners

- Remount a running filesystem
`mount -o remount,[new options] /`
- Rename all .JPG files to .jpg
`for i in *.JPG ; do \
mv $i `basename $i JPG`.jpg ; \
done`
- Rename all .JPG files to .jpg, deux
`for i in *.JPG ; do \
mv $i ${i%.JPG}.jpg ; \
done`

“Stupid Sysadmin Tricks”

One-liners

- Remove a file whose name starts with a hyphen:

```
rm -- -file
```

```
rm ./-file
```

- In-place search and replace

```
perl -pi -e 's/foo/bar/g' *.txt
```

- In-place search and replace with recursive find

```
find . -type f -print0 | \  
perl -pi -e 's/foo/bar/g'
```

“Stupid Sysadmin Tricks”

One-liners

- Find files older than (e.g.) 10 days
`find . -type f -mtime -10`
- Find files with the setuid bit on
`find . -type f -perm +4000`
`find . -type f -perm /4000`
- Find files larger than 100 MB
`find . -type f -size +100M`

Finally, a quiz

Finally, a quiz

```
#!/bin/sh
PATH=/usr/local/bin:$PATH
fqdn=$1
type=$2
[ -z "$type" ] && type=a

echo $fqdn | \
perl -pi -e 's/^\.*?([\^\.]+\.[^\.]*)$/$1/g' | \
xargs dnsqr ns | \
grep ^answer: | \
awk '{print $5}' | \
sort -u | \
while read s ; do \
    echo "$s: " ; \
    dnsq $type $fqdn $s | \
        grep ^answer: | \
        sed -e "s/^answer:/" | \
        sort -n -k4 ; \
done
```

<http://www.tldp.org/LDP/abs/html/>