Secur

Mathieu DECORE

October 29, 2003

Contents

1 Optimisation

1.1 atime and suid

Linux records information about when files were created, last modified and last accessed. There is a cost associated with recording the last access time, so not recording last access time may lead to significant performance improvements on often accessed frequently changing files such as the contents of the `/var/spool/mail`, `/var/spool/news`, HTTP and cache directories:

```
# chattr -R +A /var/spool/news
# chattr -R +A /var/spool/mail
```

You can also put in `/etc/fstab` `noatime` and make `mount -oremount /chroot/` and `cat /proc/mounts` to check if everything is ok.

See also: `atime.sh`.

Put in `/etc/fstab`:

- **nosuid** means do not allow set-user-identifier or set-group-identifier bits to take effect,
- **nodev**, do not interpret character or block special devices on this file system partition,
- **noexec**, do not allow execution of any binaries on the mounted file system

For `/home` and `/tmp` directories:

```
/dev/sda11         /tmp            ext2    defaults,rw,nosuid,nodev,noexec 1 2
/dev/sda6          /home           ext2    defaults,rw,nosuid,nodev        1 2
```

and remount this partitions:

```
# mount -oremount /home/
# mount -oremount /tmp/
```

See also: `fstab.sh`.

1.2 Kernel

1.2.1 Virtual memory

Edit `/proc/sys/vm/bdflush` to see:

1. **100** the maximum number of dirty buffers in the buffer cache. Setting this to a high value means that Linux can delay disk writes for a long time, but it also means that it will have to do a lot of I/O at once when memory becomes short ;
2. **1200** this gives the maximum number of dirty buffers that bdflush can write to the disk in one time. A high value will mean delayed, bursty I/O, while a small value can lead to memory shortage when bdflush isn't woken up often enough ;
3. **128** this is the number of buffers that bdflush will add to the list of free buffers when *`refill_freelist()`* is called. The higher the number, the more memory will be wasted and the less often *`refill_freelist()`* will need to run ;
4. **512** when this comes across more than *`nref_dirt`* dirty buffers, it will wake up bdflush.
5. unused ;
6. **5000** the *`age_buffer`* parameter govern the maximum time Linux waits before writing out a dirty buffer of data blocks to disk ;
7. **500** the *`age_super`* parameters govern the maximum time Linux waits before writing out a dirty buffer of system metadata to disk ;
8. unused ;
9. unused.

Edit `/proc/sys/vm/buffermem` to see:

1. use a minimum of `x` percent of memory for the buffer cache.

   We have *`MaxMem`* Mo memory (`free -m`, line *`Mem:`*), and we need at least *`FreeMem`* Mo of free memory (`free -m`, line *`buffers/cache`*), so:

$$MaxMem - MaxMem \times \frac{x}{100} = FreeMem$$

$$x = \frac{MaxMem - FreeMem \times 100}{MaxMem}$$

2. unused ;

3. unused.

See also: `bdflush.sh`.

1.2.2 Files

- `/proc/sys/fs/file-max` sets the maximum number of file-handles that the Linux kernel will allocate (256 for every 4M of RAM we have).

  We have ***`MaxMem`*** Mo memory (`free -m`, line ***`Mem:`***), so we can open:

  $$\frac{MaxMem}{4} \times 256 = MaxMem \times 64$$

  files.

- `/proc/sys/fs/inode-max` set the maximum number of inodes handlers. 4 times the number of opened files (at least one per file, more for large files) is a good value.

- to improve performance, we can safely set the limit of processes for the super-user root to be unlimited:

  ```
  # echo "ulimit -u unlimited" >> /root/.bashrc
  ```

- increases the system limit on open files to **90000** for the **root** account:

  ```
  # echo "ulimit -n 90000" >> /root/.bashrc
  ulimit -n 90000
  ```

See also: `file.sh`.

1.3 Limit resources for the users

The limits.conf file located under the /etc/security directory can be used to control and limit resources for the users on your system, so they can't perform denial of service attacks number of processes, amount of memory, etc. Put in /etc/security/limits.conf:

```
*        hard    core            0
*        hard    rss             5000
*        hard    nproc   20
```

This says to prohibit the creation of core files (core 0), restrict the number of processes to 20 (nproc 20), and restrict memory usage to 5M (rss 5000) for everyone except the super user **root**. You must also edit the /etc/pam.d/login file and add the following line to the bottom of the file:

```
session  required  /lib/security/pam_limits.so
```

Finally edit the /etc/profile file and change the following line:

```
ulimit -S -c 1000000 > /dev/null 2<&1
```

This modification is required so as to avoid getting error messages like this: Unable to reach limit during login:.

See also: limits.sh.

1.4 To put in /etc/profile

1.4.1 Compilation

For **i686** processor type:

```
CFLAGS='-O9 -funroll-loops -ffast-math -mcpu=pentiumpro -fomit-frame-pointer
```

For **i586** processor type:

```
CFLAGS='-O3 -funroll-loops -ffast-math -mcpu=pentium -fomit-frame-pointer -fforc
```

For others:

```
CFLAGS='-O3 -funroll-all-loops -mcpu=i486 -fomit-frame-pointer
```

See also: cflags.sh.

1.4.2 History

```
export HISTFILESIZE=20
export HISTSIZE=20
echo "rm -f \$HOME/.bash_history" >> /etc/skel/.bash_logout
```

See also: `history.sh`.

2 Network

2.1 inetd

`inetd`, called the super server, will load a network program based upon a request from the network. The `inetd.conf` file tells `inetd` which ports to listen to and what server to start for each port. The first thing to look at as soon as you put your Linux system on ANY network is what services you need to offer. Services that you do not need to offer should be disabled and uninstalled so that you have one less thing to worry about, and attackers have one less place to look for a hole.

Look at your `/etc/inetd.conf` file to see what services are being offered by your inetd program. Disable what you do not need by commenting them out by adding a # at the beginning of the line, and then sending your `inetd` process a **SIGHUP** command to update it to the current `inetd.conf` file.

Services to comment: `ftp telnet shell login exec talk ntalk imap pop-2 pop-3 finger auth`.

And then:

```
# killall -HUP inetd
```

See also: `inetd.sh`.

2.2 Better manage your TCP/IP resources

Edit:

- `/proc/sys/net/ipv4/ip_local_port_range`: defines the local port range that is used by TCP and UDP traffic to choose the local port. For high-usage systems change this to **32768 61000** ;
- `/proc/sys/net/ipv4/tcp_fin_timeout`: make the time default values for TCP/IP connection lower so that more connections can be handled by time on your TCP/IP protocol. The following will decrease the

amount of time your Linux box will try take to finish closing a connection and the amount of time before it will kill a stale connection. This will also turn off some IP extensions that aren't needed. Put **30** ;

- `/proc/sys/net/ipv4/tcp_keepalive_time`: how often TCP sends out keep alive messages, when keep alive is enabled. The default is 2 hours. Put **1800** ;

- `/proc/sys/net/ipv4/tcp_window_scaling`: enable window scaling as defined in RFC 1323. Put **0** ;

- `/proc/sys/net/ipv4/tcp_sack`: enable select acknowledgments after RFC 2018. Put **0** ;

- `/proc/sys/net/ipv4/tcp_timestamps`: enable timestamps as defined in RFC 1323. Put **0**.

See also: `tcp.sh`, `tcp-extension.sh`.

3 Security

3.1 Set minimun password length

Put in `/etc/login.defs` file:

```
PASS_MIN_LEN    8
```

to set minimun password length to **8** charaters.
See also: `login.sh`.

3.2 bash shell automatically logout after not being used for a period of time

For bash shell automatically logout after not being used for a period of time, set *`TMOUT`* variable to the number of seconds in `/etc/profile` file:

```
export TMOUT=7200
```

See also: `timeout.sh`.

3.3 Kernel security

Edit:

- `/proc/sys/net/ipv4/icmp_echo_ignore_all`: turn on (1) or off (0), if the kernel should ignore all ICMP ECHO requests, or just those to broadcast and multicast addresses. Please note that if you accept ICMP echo requests with a broadcast/multicast destination address your network may be used as an exploder for denial of service packet flooding attacks to other hosts ;

- `/proc/sys/net/ipv4/icmp_echo_ignore_broadcasts`: when a packet is sent to an IP broadcast address (i.e. 192.168.1.255) from a machine on the local network, that packet is delivered to all machines on that network. Then all the machines on a network respond to this ICMP echo request and the result can be severe network congestion or outages -denial-of-service attacks. See the RFC 2644 for more information ;

- `/proc/sys/net/ipv4/conf/default/accept_source_route`: should source routed packages be accepted or declined. The default is dependent on the kernel configuration. It's 'yes' for routers and 'no' for hosts. The IP source routing, where an IP packet contains details of the path to its intended destination, is dangerous because according to RFC 1122 the destination host must respond along the same path. If an attacker was able to send a source routed packet into your network, then he would be able to intercept the replies and fool your host into thinking it is communicating with a trusted host ;

- `/proc/sys/net/ipv4/tcp_syncookies`: only valid when the kernel was compiled with `CONFIG_SYNCOOKIES`. Send out syncookies when the syn backlog queue of a socket overflows. This is to prevent against the common 'syn flood attack'. Disabled by default. A SYN Attack is a denial of service DoS attack that consumes all the resources on your machine, forcing you to reboot ;

- `/proc/sys/net/ipv4/conf/default/accept_redirects`: this switch decides if the kernel accepts ICMP redirect messages or not. The default is 'yes', if the kernel is configured for a regular host; and 'no' for a router configuration. When hosts use a non-optimal or defunct route to a particular destination, an ICMP redirect packet is used by routers to inform the hosts what the correct route should be. If an attacker is able to forge ICMP redirect packets, he or she can alter the routing tables on the host and possibly subvert the security of the host by causing traffic to flow via a path you didn't intend ;

- `/proc/sys/net/ipv4/ip_always_defrag`: replaces the former Kernel-Configuration option `CONFIG_IP_ALWAYS_DEFRAG`. All incoming fragments (parts of IP packets that arose when some host between origin and destination decided that the packets were too large and cut them into pieces) will be reassembled (defragmented) before being processed, even if they are about to be forwarded. Only say Y here if running either a firewall that is the sole link to your network or a transparent proxy; never ever say Y here for a normal router or host. This is automagically enabled when enabling masquerading. This protection must be enabled if you use your Linux server as a gateway to masquerade internal traffic to the Internet IP Masquerading ;

- `/proc/sys/net/ipv4/icmp_ignore_bogus_error_responses`: some routers violate RFC 1122 by sending bogus responses to broadcast frames. Such violations are normally logged via a kernel warning. If this is set to TRUE, the kernel will not give such warnings, which will avoid log file clutter ;

- `/proc/sys/net/ipv4/conf/default/rp_filter`: Integer value deciding if source validation should be made. 1 means yes, 0 means no. Disabled by default, but local/broadcast address spoofing is always on. If you set this to 1 on a router that is the only connection for a network to the net , it evidently prevents spoofing attacks (forged communications that are often used in DoS attacks) against your internal networks (external addresses can still be spoofed), without the need for additional firewall rules. NOTE: this option is turned on per default only when ip_forwarding is on. For non-forwarding hosts it doesn't make much sense and makes some legal multihoming configurations impossible ;

- `/proc/sys/net/ipv4/conf/default/log_martians`: log packets with impossible source addresses (no known route) to kernel log, ie. all Spoofed Packets, Source Routed Packets, and Redirect Packets.

See also: `secur.sh`.

3.4 Allow members of group whell to execute `su`

To allow only members of group **whell** to execute `su`, add in `/etc/pam.d/su` file:

```
auth    sufficient    /lib/security/pam_rootok.so    debug
auth    required      /lib/security/pam_wheel.so     group=wheel
```

and add a user for group **wheel** (**admin** in the following example):

```
# usermod -Gwheel admin
```

See also: `su.sh`.

3.5 Delete unused groups and users

Delete the following groups if you don't use them: `adm lp news uucp games dip pppusers popusers slipusers`.
Delete `games` if you don't use X Window Server, `popusers` if you don't use pop server for email.

Delete the following users if you don't use them: `adm lp sync shutdown halt news uucp operator games gopher ftp`.
Delete `games` if you don't use X Window Server, `ftp` if you don't use FTP anonymous server.

3.6 Delete sticky bit on some programs

You can delete sticky bit on the following programs: `/usr/bin/chage /usr/bin/gpasswd /usr/bin/wall /usr/bin/chfn /usr/bin/chsh /usr/bin/newgrp /usr/bin/write /usr/sbin/usernetctl /usr/sbin/traceroute /bin/mount /bin/umount /bin/ping /sbin/netreport`. Just type:

```
chmod a-s /usr/bin/chage
```

for example.
See also: `sticky.sh`.

3.7 Logging connections to the printer

For logging of all `telnet`, `mail`, boot messages and `ssh` connections from your server to the printer attached to this server, put in `/etc/syslog.conf` file:

```
authpriv.*;mail.*;local7.*;auth.*;daemon.info /dev/lp0
```

and type:

```
# /etc/rc.d/init.d/syslog restart
```

See also: `syslog.sh`.

3.8 Lilo

Add:

```
restricted
password=<pass>
```

in `/etc/lilo.conf` to ask password if boot with parameters (eg. `linux single`).

And then:

```
# /sbin/lilo
```

See also: `lilo.sh`.

References

[1] **Securing and Optimizing Linux**, Gerhard Mourani, 2000.
`http://www.linuxdoc.org/`