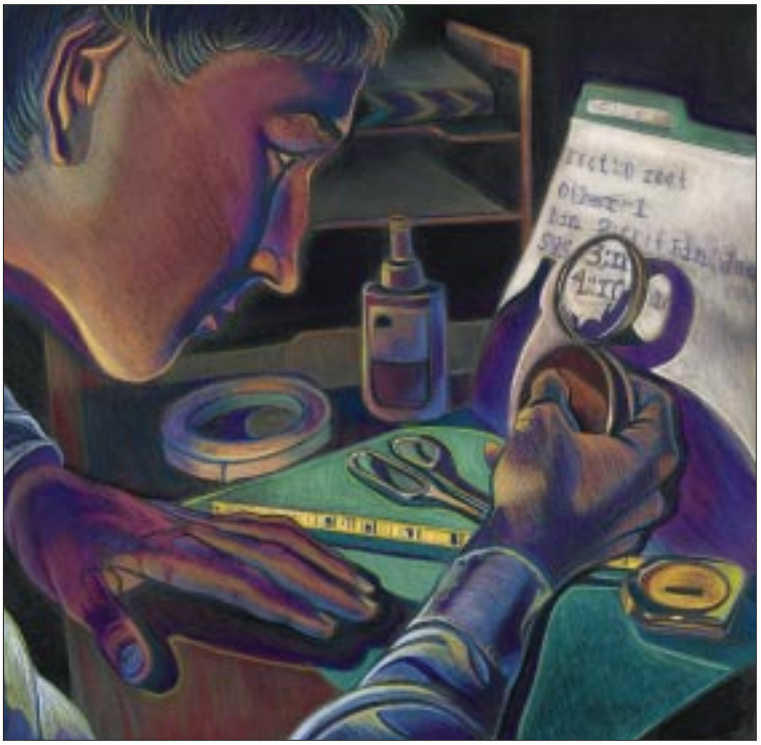
by S. Lee Henry



KRISTINA WEISS

Checking Groups

Everybody and his dog knows how to set up a group in UNIX. The `/etc/group` file allows groups to be assigned textual names, passwords (rarely used) and members. But everybody and his dog doesn't know, at least myself and my dog just found out, that there are some problems that can crop up in the `/etc/group` file, and that there is a neat little tool for helping to detect them.

The group check tool, `grpck`, checks the format of the group file. It tells you if any names are redundantly defined, that is, if they are members of a UNIX group by virtue of the `GID` field in the `etc/passwd` entries and also included in the `/etc/group` file.

Although being doubly defined as members of a group may not cause problems for those particular users, it might cause problems for other users of that group. For example, once a group entry gets to be longer than *X* members or *Y* characters long, the line in the `/etc/group` file will be declared too long by the `grpck` tool. As a result, users at the tail end of the group may not be given the group privileges that you expect.

If you run the `grpck` command against the `/etc/group` file (what other file would you check?), it will inform you of users who are already members of a given group and if a group definition is too long. It will also tell you if any members of a group don't exist in the `/etc/passwd` and `/etc/shadow` files (that is, that they are not defined on the system). This can help you pinpoint and remove old usernames from your `/etc/group` file. You will see errors like the one shown below if you have problems.

If you find that you have a large number of doubly defined users, you can remove them from the group corresponding to their `GID` entries. The script shown in Figure 1 will write out a group (of your choosing) to a file and then check each member against the appropriate `/etc/passwd` entry.

At the end, the script will display a complete group entry in sorted order with any doubly defined usernames removed. You can cut and paste this on top of the old `/etc/group` entry, or you can remove the old entry and insert the new as shown in Figure 2. Removing the doubly defined members will reduce the overall size of the group entry and may bring it back within the allowable group size.

I couldn't find in any of the `man` pages I read what the maximum allowable size of a group defined within the `/etc/group` file is or any hints that it might be possible to include groups as members of groups (as I can with

```
timmy - Duplicate logname entry (gid first occurs in passwd entry)
Line too long
```

Figure 1. Checking Group Members Against /etc/passwd Entries

```
#!/bin/csh
#

echo -n "Please enter name of group> "
set GRPNAME = $<
set GRPNO = `grep ^$GRPNAME":" /etc/group | awk -F: '{print $3}'`
if ("$GRPNO" == "") then
       echo "Sorry -- there is no such group"
       return
else
       echo "Checking group $GRPNAME, group number $GRPNO"
endif

foreach person (`grep ^$GRPNAME":" /etc/group | tr ":," "\012\012" | tail +4`)
       set PRIME_GRP = `grep ^$person":" /etc/passwd | awk -F: '{print $4}'`
       if ($PRIME_GRP != "$GRPNO") then
              touch /tmp/grp$GRPNO.outsiders$$
              echo $person >> /tmp/grp$GRPNO.outsiders$$
       endif
end

echo "Here are the usernames which NEED to be in the $GRPNAME group:"
set NEWGRP = `cat /tmp/grp$GRPNO.outsiders$$ | sort | tr "\012" "," | awk '{print
substr($0,1,length($0)-1)}'`
echo $NEWGRP
rm /tmp/grp$GRPNO.outsiders$$
```

Figure 2. Inserting a New /etc/group File

```
myhost# cp -p /etc/group /etc/group-
myhost# cat /etc/group | grep v :30: > /tmp/group
myhost# fix_group
Which group?> sales
sales::30:billybob,corey,nici,timmy,vancouver
myhost# echo sales::30:billybob,corey,nici,timmy,vancouver >> /tmp/group
myhost# mv /tmp/group /etc/group
```

Figure 3. Figuring out the Limit

```
head -80 /etc/passwd | awk -F: '{print length($1),$1}' | sort -n | awk {print $2} | tr \012 ,
tail -60 /etc/passwd | awk -F: '{print length($1),$1}' | sort -n | awk {print $2} | tr \012 ,
```

netgroups). Figuring out the limit, therefore, took a bit of experimentation. Here's what I did: First, I made a list of users taken from the top of the list (the first 80) and another list of users from the bottom of the list (the bottom 60); second, I made groups out of them (commands are shown in Figure 3), this left me with one large group composed of a lot of short usernames and one large group with fewer longer names.

I simply removed the final comma and added these lines to my `/etc/group` file with the strings `sales1::66:` and `sales2::77:` preceding them. Then, I repeatedly used the `grpck` command and dropped members from the tail end of the first group and the head end of the other until each was acceptable–that is, until `grpck` no longer complained.

The result? The list with many users with short usernames wound up with 75 members and a total length of 499. The list with fewer users with longer usernames wound up with 47 members and a total length of 497. Clearly, this suggests that the limiting factor is the length of the group

Figure 4. Showing the Length and Number of Group Members

```
# grep sales /etc/group | awk -F, '{print NF}'
75
47
# grep sales /etc/group | awk -F: '{print length($4)}'
499
497
```


record. To show the length and number of members in your groups, use the commands shown in Figure 4.

The `/etc/group` file is only one way to define groups, of course. NIS and NIS+ may have different limitations for group members.

Another problem that Jaspar and I ran into with groups is related to the length of usernames in UNIX. Most places I've worked have limited usernames to eight characters to avoid some inconsistencies that seem to crop up with longer usernames. It seems to me that UNIX, in some cases, ignores characters after the eighth (as it does with passwords). Then, at other times, it pays attention to all of the characters in a username. We noticed that truncated usernames had crept into our `/etc/group` file when these users weren't being treated as members of the group. The `grpck` tool finds these problems easily, issuing a "Logname not found in `passwd` file" error message.

With my groups properly defined, I can avoid overusing world privilege. As far as I can tell, any number of users can be defined as members of the same group if the assignment is made in the `/etc/passwd` file. It is only when I want a large number of the same users to be members of a second group that I run into problems.

S. Lee Henry *is a security services engineer at Infonet in El Segundo, CA, where no one else necessarily shares any of her opinions. Jaspar chases cats for a living and actually knows very little about UNIX.*