

Re: [patch] rm can have undesired side-effects

Source: <http://unix.derkeiler.com/Mailing-Lists/FreeBSD/hackers/2006-10/msg00258.html>

- *From:* Bakul Shah <bakul@xxxxxxxxxxxxxx>
 - *Date:* Mon, 30 Oct 2006 11:48:44 -0800
-

Doug Barton writes:

Bakul Shah wrote:

Sorry if I tuned in late:-)

I vote for taking **out** -P. It is an ill-designed feature. Or if you keep it, also add it to mv, cp -f & ln -f since these commands can also unlink a file and once unlinked in this matter you can't scrub it. And also fix up the behavior for -P when multiple links. And since mv can use rename(2), you will have to also dirty up the kernel interface somehow. Not to mention even editing such a sensitive file can leave stuff all over the disk that a bad guy can get at. If you are truly paranoid (as opposed to paranoid only when on meds) you know how bad that is!

If you are that concious about scrubbing why not add scrubbing as a mount option (suggested option: -o paranoid) then at least it will be handled consistently.

The patches to implement your suggestions didn't make it through on this message. Please feel free to post them for review and send the URL to the list.

Writing code is the easy part, too easy in fact, which is part of the problem. Interface changes need to be discussed and made carefully. But since you asked, here's the patch to remove -P from rm.

Index: rm.c

```
=====
RCS file: /home/ncvs/src/bin/rm/rm.c,v
retrieving revision 1.54
diff -w -u -b -r1.54 rm.c
--- rm.c 15 Apr 2006 09:26:23 -0000 1.54
+++ rm.c 30 Oct 2006 19:43:40 -0000
```

Re: [patch] rm can have undesired side-effects

```
@@ -57,7 +57,11 @@
#include <sysexits.h>
#include <unistd.h>

+#ifdef HALF_PARANOID
int dflag, eval, fflag, iflag, Pflag, vflag, Wflag, stdin_ok;
+#else
+int dflag, eval, fflag, iflag, vflag, Wflag, stdin_ok;
+#endif
int rflag, Iflag;
uid_t uid;
```

```
@@ -66,7 +70,9 @@
void checkdot(char **);
void checkslash(char **);
void rm_file(char **);
+#ifdef HALF_PARANOID
int rm_overwrite(char *, struct stat *);
+#endif
void rm_tree(char **);
void usage(void);
```

```
@@ -103,8 +109,13 @@
exit(eval);
}
```

```
+#ifdef HALF_PARANOID
Pflag = rflag = 0;
while ((ch = getopt(argc, argv, "dfiIPrvW")) != -1)
+#else
+ rflag = 0;
+ while ((ch = getopt(argc, argv, "dfiIRrvW")) != -1)
+#endif
switch(ch) {
case 'd':
dflag = 1;
@@ -120,9 +131,11 @@
case 'I':
Iflag = 1;
break;
+#ifdef HALF_PARANOID
case 'P':
Pflag = 1;
break;
+#endif
case 'R':
case 'r': /* Compatibility. */
rflag = 1;
@@ -289,9 +302,11 @@
continue;
/* FALLTHROUGH */
```

Re: [patch] rm can have undesired side-effects

```
default:
+ifdef HALF_PARANOID
if (Pflag)
if (!rm_overwrite(p->fts_accpath, NULL))
continue;
+endif
rval = unlink(p->fts_accpath);
if (rval == 0 || (fflag && errno == ENOENT)) {
if (rval == 0 && vflag)
@@ -357,9 +372,11 @@
else if (S_ISDIR(sb.st_mode))
rval = rmdir(f);
else {
+ifdef HALF_PARANOID
if (Pflag)
if (!rm_overwrite(f, &sb))
continue;
+endif
rval = unlink(f);
}
}
@@ -372,6 +389,7 @@
}
}

+ifdef HALF_PARANOID
/*
* rm_overwrite --
* Overwrite the file 3 times with varying bit patterns.
@@ -436,7 +454,7 @@
warn("%s", file);
return (0);
}
-
+endif

int
check(char *path, char *name, struct stat *sp)
@@ -462,6 +480,7 @@
strmode(sp->st_mode, modep);
if ((flagsp = fflagstostr(sp->st_flags)) == NULL)
err(1, "fflagstostr");
+ifdef HALF_PARANOID
if (Pflag)
errx(1,
"%s: -P was specified, but file is not writable",
@@ -472,6 +491,7 @@
group_from_gid(sp->st_gid, 0),
*flagsp ? flagsp : "", *flagsp ? " " : "",
path);
+endif
```

Re: [patch] rm can have undesired side-effects

```
free(flagsp);
}
(void)fflush(stderr);
@@ -583,7 +603,11 @@
{

(void)fprintf(stderr, "%s\n%s\n",
#ifdef HALF_PARANOID
"usage: rm [-f | -i] [-dIPRrvW] file ...",
#else
+ "usage: rm [-f | -i] [-dIRrvW] file ...",
#endif
" unlink file");
exit(EX_USAGE);
}
```

freebsd-hackers@xxxxxxxxxxx mailing list

<http://lists.freebsd.org/mailman/listinfo/freebsd-hackers>

To unsubscribe, send any mail to "freebsd-hackers-unsubscribe@xxxxxxxxxxx"