Rachel Crofts, Scotsman, 2004.11.29:

“Consumers warned of ‘criminal cashback’ scam

“Police today warned consumers to be on their guard against a new type of fraud being dubbed ‘criminal cashback.’ . . .

“The targeted victim will be contacted by the fraudster posing as a buyer. Whatever the price of the item for sale, the ‘buyer’ or an associate will send a [check] for significantly more than the asking price.

“The ‘buyer’ will then enter into an agreement with the vendor that this overpayment will be returned to them via money transfer . . .

“If the [check] is fraudulent or stolen its value will be taken back out of the account to which it was paid when this fact is discovered. This can be up to weeks later.”

The /tmp-cleaning problem

Widespread bug for many years:

```
    rm -f \
    ‘find /tmp -mtime +7 -print‘
```

Alternate version, same bug:

```
    find /tmp -mtime +7 -print \
    | xargs rm -f
```

root runs this every day.
Idea: Remove old files in /tmp.

Impact: Any local user can
delete any file on the system.

```
find /tmp -mtime +7 -print
```

looks through files in `/tmp`,
in subdirectories of `/tmp`,
in subsubdirectories, etc.,
for files not modified
in the past 7 days.

It prints
name of first file, newline,
name of second file, newline,
etc.

rm -f ‘...‘
takes output of ...;
splits it into strings
separated by space, tab, newline;
and runs rm -f
with those strings as arguments.

e.g. If /tmp has old files
/tmp/foo and /tmp/bar:
find /tmp ... prints
"/tmp/foo\n/tmp/bar\n".
The shell runs
"rm","-f","/tmp/foo","/tmp/bar".
rm removes /tmp/foo
and then /tmp/bar.

Minor bug: There may be too many files to fit on `rm` command line; `execve` limits the length of a command line.

Fix: Change `rm -f ‘...‘` to `... | xargs rm -f`. `xargs` runs `rm -f` several times if necessary, breaking command line into several pieces that fit below length limit.

Major bug: The transformation
from string to list of names
doesn't undo the transformation
from list of names to string.

Attacker creates
directory `"/tmp/x "`;
directory `"/tmp/x /etc"`;
file `"/tmp/x /etc/passwd"`.

`find` prints
`"/tmp/x /etc/passwd\n"`.
Shell (or `xargs`) runs
`"rm","-f","/tmp/x","/etc/passwd"`.
`rm` removes `/etc/passwd`.

Fix 1:

```
    find /tmp -mtime +7 -print0 \
    | xargs -0 rm -f
```

`find ... -print0` uses byte 0
to separate filenames:
`"/tmp/x /etc/passwd\0/tmp/bar\0"`.

`xargs -0` understands perfectly,
looking for byte 0
as the only separator.

Filenames cannot contain byte 0,
so no possibility of error.

Fix 2:

```
    find /tmp -mtime +7 \
    -exec rm -f '{}' ';'
```

runs `rm -f` separately
for each file to be removed.

```
"rm","-f","/tmp/x /etc/passwd";
"rm","-f","/tmp/bar";
```

etc.

Slow, but who cares?

The bad news: Both fixes still allow any local user to delete any file on the system.

Unsuccessful attack:
Attacker runs

```
    ln -s / /tmp/x
```

to create /tmp/x as a symbolic link to /.

Sysadmin intended to remove only files in /tmp; but now /tmp/x/etc/passwd is another name for /etc/passwd.

Fortunately for sysadmin, find skips symlinks.

Successful attack using TOCTOU gap:
Attacker creates
directory `/tmp/x`,
directory `/tmp/x/etc`,
file `/tmp/x/etc/passwd`.

`find` discovers file,
prints name `/tmp/x/etc/passwd`.

Attacker quickly renames
`/tmp/x` as `/tmp/x2`,
symlinks `/tmp/x` to `/`.

`rm -f /tmp/x/etc/passwd`
now removes `/etc/passwd`.

Complicated fix:
set process working directory
to the directory containing
the file to be removed;
remove file using non-/ name.
BSD `find ... -delete` does this.

Much better fix:
Stop using `/tmp`.
Have separate `/home/joe/tmp`,
cleaned by a `joe` process.