The Age, Australia, 2004.10.29:

“Thieves caught driving stolen Lamborghini

“A bright orange $415,000 Lamborghini bought from a Sydney car dealer with a fraudulent cheque was on the road for less than a day before police pulled it over.”

The setuid-open-exec security problem

Sendmail, setuid, opens various
`root` files for writing.
Process fds 3, 4, 5, etc.

Sendmail then runs `/home/joe/evil`.
Process still has the `root` files
open for writing: fds 3, 4, 5, etc.

`/home/joe/evil` then writes to files:
e.g., `write(4,...)`.

Fix 1: Sendmail closes fds 3, 4, 5, etc.
before `execve`. Difficulty: no syscall
reveals exactly which fds are open.

Fix 2: Each fd has close-on-exec flag.
Sendmail sets close-on-exec flag
for each file it opens.

The setuid-open-stderr security problem

Consider a password-changing program:

1. Open system's password file
   for reading and writing.
2. Find user's old password.
3. Prompt user for old password,
   by writing to `stderr`.
4. Read line from user.
5. Stop if the line doesn't
   match the old password.
6. Prompt user for new password.
7. Read line from user.
8. Change password in file.

This program is setuid so that
it can access the password file.

Joe runs `/home/joe/evil`,
which calls `close(2)` and
runs the password-changing program.
What happens?

Program opens password file.
`open()` uses descriptor 2.

Program then writes to `stderr`,
which is descriptor 2. Oops!

Impact: depends on exactly what
program writes. Maybe Joe can
control other users' passwords;
maybe just destroy passwords.

Setuid programs can't trust fds.