Setuid

Some legitimate actions require more privileges than UID
e.g. how users change their passwords stored in root-owned `/etc/passwd` and `/etc/shadow` files?

➡ Solution - `setuid` and `setgid` programs

- Run with privileges of file's owner or group
- Each process has real and effective UID/GID
- Real is user who launched `setuid` program
- Effective is owner/group of file, used in access checks

Shown as "s" in file listings

```
-rws--x--x 1 root root 52528 Oct 29 08:54 /bin/passwd
```

- Obviously need to own file to set the `setuid` bit
- Need to own file and be in group to set `setgid` bit

Setuid

Examples

- `passwd` – changes user's password
- `su` – acquire new user ID (given correct password)
- `sudo` – run one command as root
- `ping` (historically) – uses raw IP sockets to send/receive ICMP

Have to be very careful when writing `setuid` code

- Attackers can run `setuid` programs any time (no need to wait for root to run a vulnerable job)
- Attacker controls many aspects of program's environment

➡ You will write such attack in CSCD27