Putting it All Together

- Let's look at some relatively simple privileged programs
- How do they combine the different mechanisms we've seen?
- What are the threats? The defenses?

The “Passwd” Command

- Permits users to change their own passwords
- In other words, controls system access
- Very security-sensitive!
- How does it work?

Necessary Files

- **`/etc/passwd`** — must be world-readable, for historical reasons
  - ☞ Maps numeric UID to/from username
- Historical format:

  `root:8.KxUJ8mGHCwq:0:0:Root:/root:/bin/sh`
- Fields: username, hashed password, numeric uid, numeric gid, name, home directory, shell
- Numeric uid/gid is what is stored for files
- Password is two bytes of salt, 11 bytes of encryption output
- Encoded in base 64 format: A-Za-z0-9./

Storing the Hashed Password

- Better not make it world-readable
- Store in a *shadow password* file
- That file can be read-protected

File Permissions

```
$ ls -l /etc/passwd /etc/shadow
-rw-r--r--   1 root  root  671 Oct  3 10:42 /etc/passwd
-r--------   1 root  root  312 Oct  3 10:42 /etc/shadow
```

Must Be Owned by Root!

- Ownership of that file is equivalent to root permissions
- Anyone who can rewrite it can give themselves root permissions
- Cannot use lesser permissions
- Note: adding a line to that file (often with a text editor) is the first step in adding a user login to the system

Implications of the Numeric UID/GUID

- Assigning a UID to a username grants access to that UID's files
- In other words, anyone with write permission on **`/etc/passwd`** has access to all files on the systm
- Consequence: even if we changed the kernel so that root didn't have direct access to all files, this mechanism provides indirect access to all files
- Conclusion: Cannot give root control over UID assignment on secure systems

What Else Shouldn't Root Be Able to Change?

- The user's password!
- Attack: change the user's password to something you know
- Windows XP does not give Administrator either such power

The Passwd Command

- Clearly, must be setUID to root
- Must be carefully written. . .

Authenticating the User

- Passwd program has real UID
- Demand old password — why?
- ☞ Guard against someone doing permanent damage with minimal access
- Root can change other user's passwords

Where Does the Salt Come From?

- Passwd command generates random number
- Need this be true-random?
- No — “probably different” will suffice.
- Seed ordinary pseudo-random number generator with time and PID

Restricting Access

- Suppose only a few people were allowed to change their own passwords
- Take away other-execute permission; put those people in the same group as “passwd”

Front Ends

- What about the help desk, for forgotten passwords?
- Have a setUID root front end that invokes passwd
- Validate: make sure they can only change certain users' passwords
- Log it! (Much more later in the semester)

Making a Temporary Copy

- Must copy password file to temporary location and back to change a password
- Watch out for race condition attacks!
- Actual solution: put temporary file in **`/etc`** instead of **`/tmp`**; avoid whole problem
- Secondary benefit: use temporary file as lock file, and as recovery location in case of crash

Update in Place

- Password changes could overwrite the file in place
- Doesn’t work for use add/delete or name change
- Still need locking

Passwords on the Command Line?

- Bad idea — **`ps`** shows it
- Bad idea — may be in shell history file

```
$ history 12
12      date
13      man setuid
14      ls -l ‘tty
```

- Your terminal isn’t readable by others:

```
$ ls -l ‘tty‘
crw--w----  1 smb tty 136, 5 Oct 26 14:24 /dev/pts/5
```

Changing Your Name

- Chsh is like passwd, but it lets you change other fields
- Ordinary users can change shell and human-readable name; root can change other fields
- *Much* more dangerous than passwd

Input Filtering

- What if user supplies new shell or name with embedded colons? Embedded newlines? Both?
- Could create fake entries!
- Must filter for such things

Features Used

- Access control
- Locking/race prevention
- Authentication
- Privilege (setUID)
- Filtering