SEED SECURITY LABS

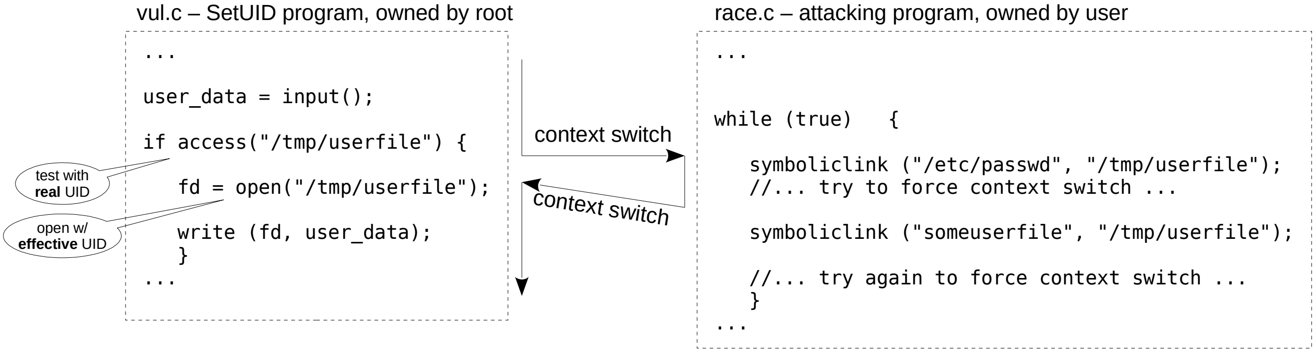
Race Condition Vulnerability Lab

General problem

- Race conditions:
  - final result depends on timing
  - solution: guarantee atomic serialization of operations
  - e.g. concurrent withdrawal of funds:
    - atomic: 1st, get balance; 2nd, withdraw possible amount
- In general, for security, guarantee:
  - atomicity of Time-Of-Check To Time-Of-Use (TOCTTOU)

Vulnerability demonstration: Unix

- any user can create files (and symbolic links) in /tmp
- use a root SETUID program that reads user file in /tmp and acts on it (e.g writes to it)
- change file in /tmp to symbolic link to system file (e.g. /etc/passwd)
  - try to do so between system testing of user file and system changing it to link
  - if this succeeds, system target will be compromised



vul.c – SetUID program, owned by root

```
...

user_data = input();

if access("/tmp/userfile") {

   fd = open("/tmp/userfile");

   write (fd, user_data);
   }
...
```

race.c – attacking program, owned by user

```
...


while (true)   {

   symboliclink ("/etc/passwd", "/tmp/userfile");
   //... try to force context switch ...

   symboliclink ("someuserfile", "/tmp/userfile");

   //... try again to force context switch ...
   }
...
```

Software bug

- CWE-61: UNIX Symbolic Link (Symlink) Following [1]
    - The software, when opening a file or directory, does not sufficiently account for when the file is a symbolic link that resolves to a target outside of the intended control sphere.
      This could allow an attacker to cause the software to operate on unauthorized files.

[1] remember: CWE - Common Weakness Enumeration «has to do with the vulnerability – not the instance within a product or system»
CVE - Common Vulnerabilities and Exposures «has to do with the specific instance within a product or system – not the underlying flaw»