```
                        _..._
                     .-'_..._''.
                   .' .'      '.\     __.....__
                  / .'            .-''         '.
.-,.--.          . '             /     .-''"'-.  `.
|  .-. |    __   | |            /     /________\   \
| |  | | .:--.'. | |            |                  |
| |  | |/ |   \ |. '            \    .-------------'
| |  '- `" __ | | \ '.          .\    '-.____...---.
| |      .'.''| |  '. `._____.-'/ `.             .'
| |     / /   | |_   `-.______ /    `''-...... -'
|_|     \ \._,\ '/            `
         `--'  `"
       _..._       .-'''-.                                                 .-'''-.
    .-'_..._''.   '   _    \             _______                          '   _    \
  .' .'      '.\/   /`  '.   \    _..._   \  ___ `'.   .--.         .--. /   /`  '.   \    _..._
 / .'          .   |     \  '  .'     '.  ' |--.\  \  |__|         |__|.   |     \  '  .'     '.
.  '            |   '      |  '.   .-.   . | |    \  ' .--.     .|  .--.|   '      |  '.   .-.   .
| |            \    \     / / |  '   '  | | |     |  '|  |   .' |_ |  |\    \     / / |  '   '  |
| |             `.   ` ..' /  |  |   |  | | |     |  ||  | .'     ||  | `.   ` ..' /  |  |   |  |       _
.  '                '-...-'`   |  |   |  | | |     ' .'|  |'--.  .-'|  |    '-...-'`   |  |   |  |     .' |
 \ '.          .              |  |   |  | | |___.' /' |  |   |  |  |  |               |  |   |  |    .   | /
  '. `._____.-'/              |  |   |  |/_______.'/  |__|   |  |  |__|               |  |   |  |  .'.'| |//
    `-.______ /               |  |   |  |\_______|/          |  '.'                   |  |   |  |.'.'.-'  /
             `                |  |   |  |                    |   /                    |  |   |  |.'   \_.'
                              '--'   '--'                    `'_'                     '--'   '--'
```

Race Condition Overview

- A situation that occurs when a 2 or more functions that are meant to occur in sequence as intended
- If there is an expectation of a particular order issues can arise when this is not the case

TOCTTOU - “TOCK too”

- Time of Check to Time of Use
- A class of bug relating to race conditions where something changes between the checking of a condition and the use of the results of that check

An Example

- A system my check a user's certificate before performing a privileged actions
- During this check several items are verified
  - Digital signature chain is valid
  - The certificate hasn't expired
  - The certificate hasn't been revoked
  - User's name/domain/organization on cert match the registered user

Can you think of a race condition?

The Race Condition

- What if these are each checked in parallel and the action they are trying to take after verification does not block until all checks are passed
- The action may occur before the system realizes the user is using a revoked cert

How would you fix this?

Mitigations

- Perform the checks sequentially
- Have a final check of the state of all checks
- Block the threads until they are all done

Another Example

- A password manager checks to see if there is a login form on the page
- If a form is found it checks to make sure the form is visible to the user
- The password manager checks for other login forms on the page
- If the form is visible it fills in the form with the user's credentials

Can you think of a race condition?

The Race Condition

- What if the malicious JS on the page covers the form right after it is checked for visibility?
- This could be a few milliseconds after the page is rendered but before a person could recognize what is going on
- The user wouldn’t see the hidden form, the password manager would assume it is visible and fill it in

How would you fix this?

Mitigations

- Don’t perform other processing between the visibility check and form fill in
- Don’t fill in the form automatically
- Hook the form and check for state change

Yet Another Example

- A banking application has a bill-pay feature
- The bill-pay system checks to make sure a user has money in their account before sending it off
- The system then logs the transaction, processes the order, and the money is wired

Can you think of a race condition?

The Race Condition

- What if a user puts in for 2 bill-pays at the same time?
- They may be able to transfer more money they actually have.
- Both bill-pay transactions would check the balance before actually processing the order

How would you fix this?

Mitigations

- Keep track of an “available balance” in addition to the “actual balance”
  - Deduct from the available balance after the initial balance check
  - Have bill-pay check the available balance and actual balance before processing
- Check the balance again after processing but before wiring

One More Example

- Some C code from Unix:

```
if (access("file", W_OK) != 0) {
   exit(1);
}

fd = open("file", O_WRONLY);
write(fd, buffer, sizeof(buffer));
```

Can you think of a race condition?

The Race Condition

- After the check overwrite the file with a sym link to another file on the system
- The check would pass but then the user would be given access to another file on the system

How would you fix this?

Mitigations

- Lock the file