CS 4530 March 18, 2021 - Security

Administrative notes:
1. HW4 due tomorrow.
Reminder: useEffect runs each time a component renders. Specify a dependency array to run the hook only the first time the component renders + whenever those dependencies change. See our extended Todo app example ' from Week 6

Q: When does a component get rendered? When any **state** is updated
Example: const [items, setItems] = useState<TodoItem[]>([]);
setItems().... // Causes a re-render

```
useEffect(() =>{
    console.log('hello');
    }
); //When is "hello" printed? - Any time the state changes

useEffect(() =>{
    console.log('hello');
    setItems(['some new todoItem']);
    }
); //When is "hello" printed? - Any time the state changes. How many times is it printed? Produces an infinite stream of "hello"

useEffect(() =>{
    console.log('hello');
    setItems(['some new todoItem']);
    },
    [setItems] // Dependencies for our useEffect
        // Only trigger effect if these dependencies change
        // Q: How many times is hello printed here?
        // Only when setItems is changed. Items might change, but setItems will not!
        // So: effectively, only called on the first render
);
```

Agenda:
1. Security review + Discussion
2. Cross site scripting + LGTM demo
3. Security testing activity

Agenda:
1. Security review + Discussion
2. Cross site scripting + LGTM demo
3. Security testing activity

Security Review

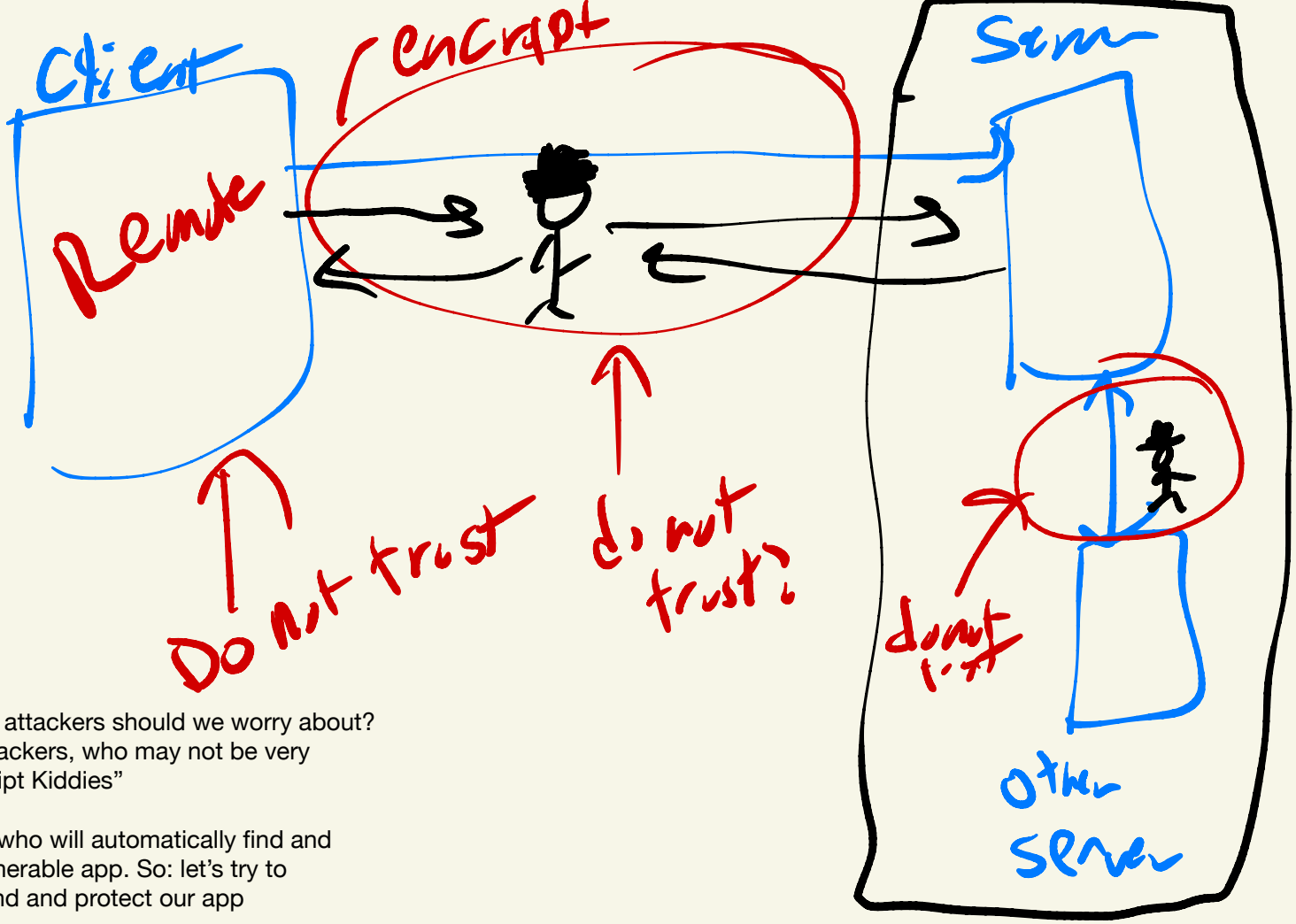
Q: Have you ever written anything that got hacked, or worked with some application that had been hacked into?
- Example: server code receives a request like: /page?content=welcome
  - Server looks for a file called welcome.txt
  - Server responds with the contents of welcome.txt
  - Client requests: /page?content=/etc/passwd
- SQL injection:
  - const queryString = 'SELECT * from table where name="' + userName + '"';
  - userName = "5' or '1'='1'"

Q: How have you thought about security when developing software in other classes, or in Co-ops?
- Take security training before starting work
  - OWASP
  - Security briefings
- Gotten advice to avoid vulnerable stuff - use SQL prepared statements, React for XSS

Q: Why build a threat model?
- Outline the potential attack points in our application, what we will do to protect, who might be protecting

Q: What are we protecting?
- Confidentiality
- Integrity
- Availability

Q: What a reasonable threat model for a web app?
- Trust:
  - People writing our code
  - Our dependencies that we import?
    - Sometimes... Maybe only if the dependency is "popular"
    - What if they are compromised? ESlint + Solarwinds
    - Eslnt <— not actually "eslint"
- Don't trust:
  - Remote user [Could be anyone]



Q: What kind of attackers should we worry about?
A: Malicious attackers, who may not be very motivated. “Script Kiddies”

AKA: someone who will automatically find and exploit your vulnerable app. So: let’s try to automatically find and protect our app