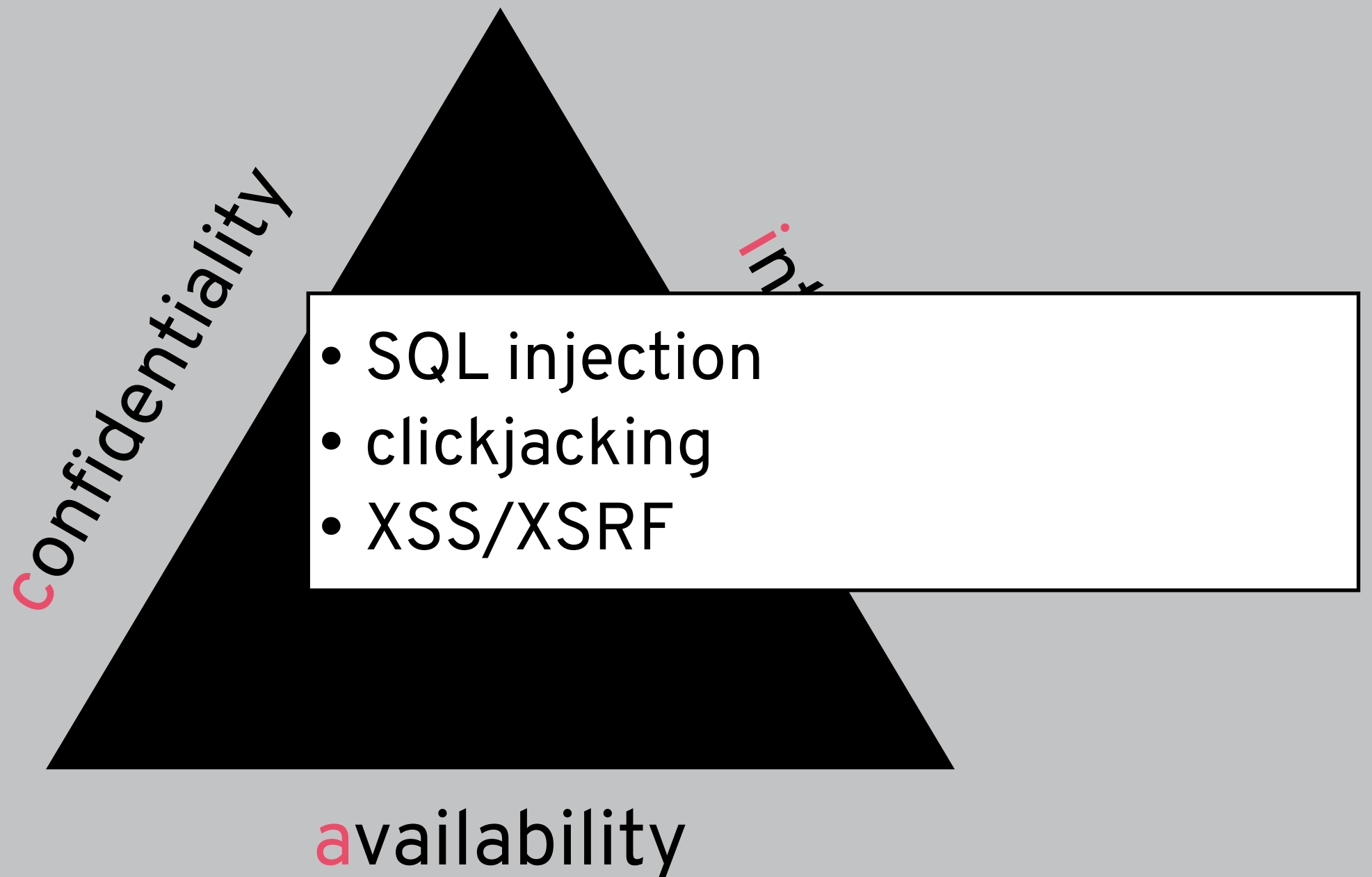
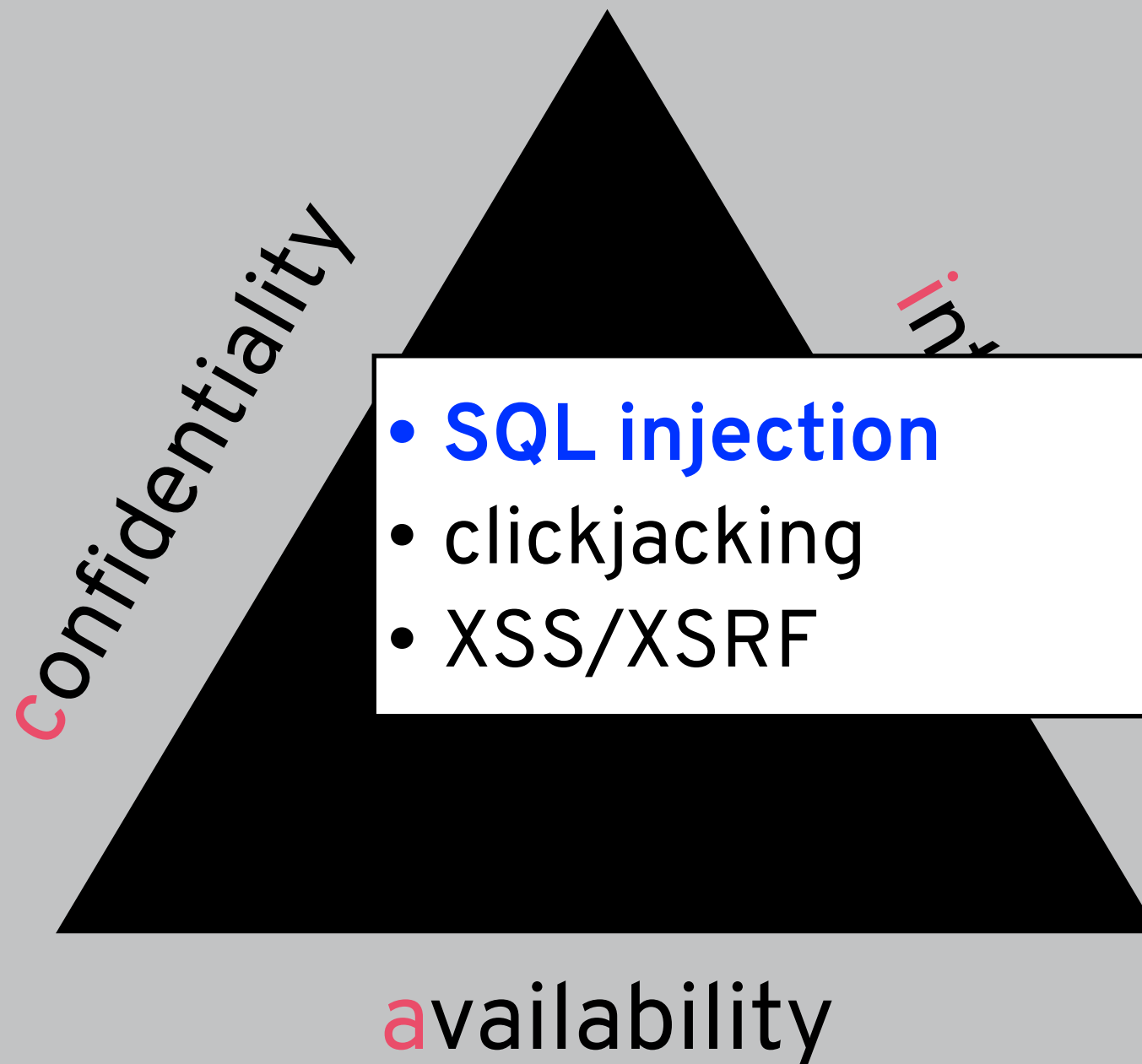

SECURITY (COMP0141): ATTACKS ON INTEGRITY



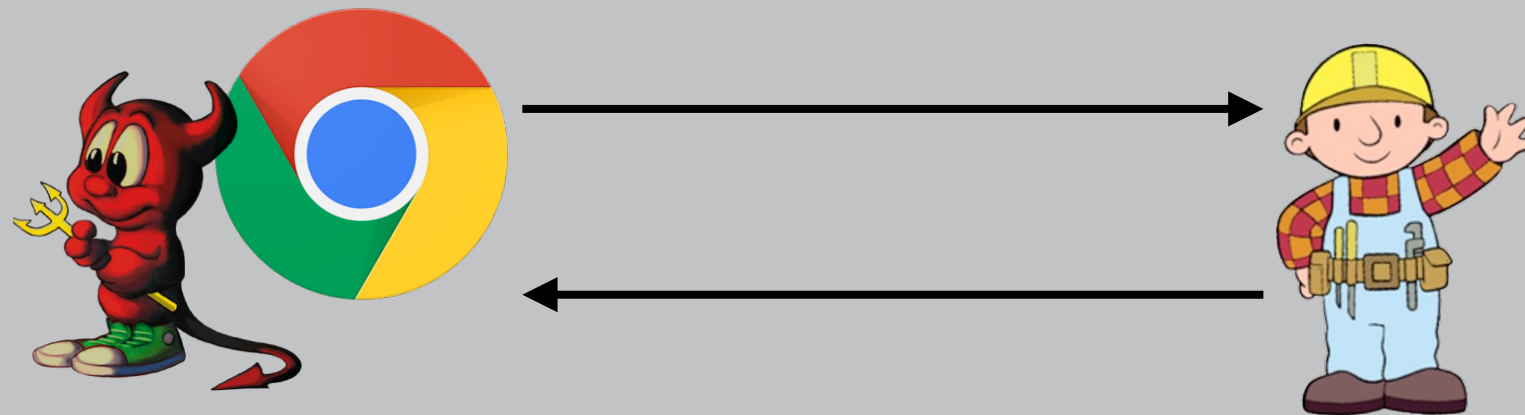
INTEGRITY, REVISITED



INTEGRITY, REVISITED



THREAT MODEL



Is the user trusted by the server? or the browser?

- SQL injection
- Click fraud

SQL BASICS

SQL = Structured Query Language

```
SELECT * FROM shop WHERE price < 10 ORDER BY type;
```

Logical expressions: AND, OR, NOT

Comment: --

Statement terminator: ;

SQL INJECTION

Server-side applications generate SQL queries based on **arbitrary user input**

```
query = "SELECT count(*) FROM users WHERE  
user_name = ' " + req.getParam("user") + " ' " + " AND  
user_pass = ' " + req.getParam("pass") + " ' " ;
```

What's the big deal?

SQL INJECTION

Server-side applications generate SQL queries based on **arbitrary user input**

```
query = "SELECT count(*) FROM users WHERE  
user_name = ' " + req.getParam("user") + " ' " + " AND  
user_pass = ' " + req.getParam("pass") + " ' " ;
```

```
"user" = alice' ;--  
query = "SELECT count(*) FROM users WHERE  
user_name = 'alice' ;-- ..."
```

```
"user" = alice  
"pass" = foo' OR 1=1 ;--  
query = "SELECT count(*) FROM users WHERE  
user_name = 'alice' AND user_pass = 'foo' OR 1=1 ;"
```

SQL INJECTION

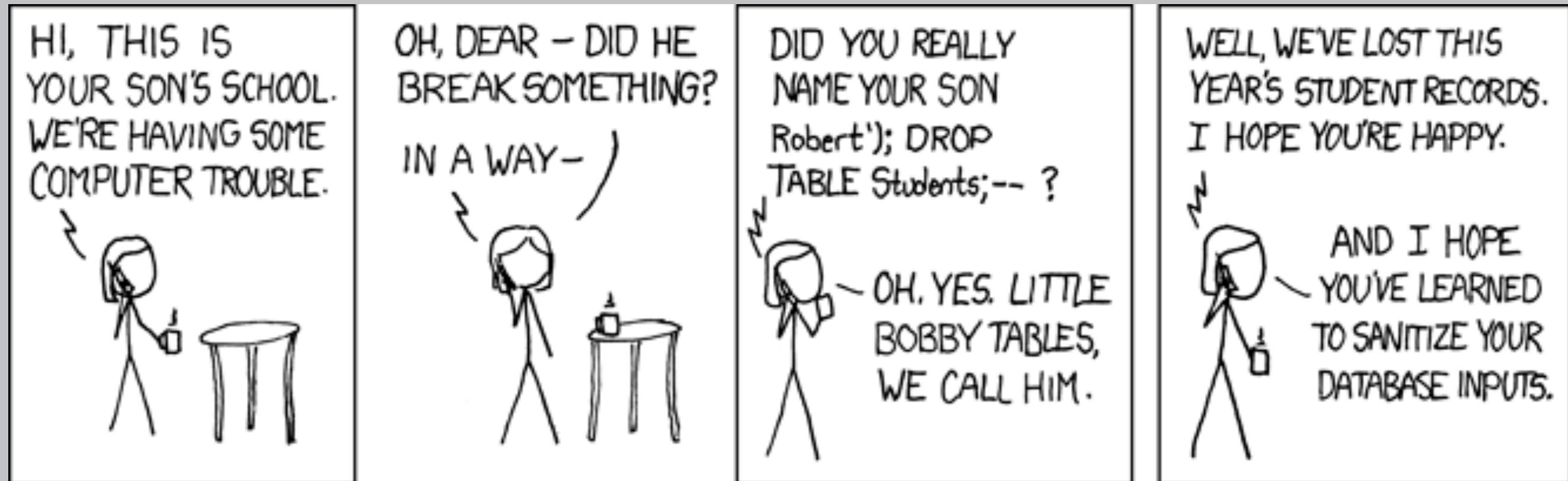
Server-side applications generate SQL queries based on **arbitrary user input**

```
query = "SELECT count(*) FROM users WHERE  
user_name = ' " + req.getParam("user") + "' " + " AND  
user_pass = ' " + req.getParam("pass") + "'";
```

More generally, can execute any SQL command

```
"user" = 'alice'; DROP TABLE users;--  
query = "SELECT count(*) FROM users WHERE  
user_name = 'alice'; DROP TABLE users;- ..."
```


LITTLE BOBBY TABLES



SQL INJECTION MITIGATIONS

Server-side applications generate SQL queries based on **arbitrary user input**

Solution? Don't accept arbitrary user input!

Parameterised queries: pre-compiled queries that separate commands from input

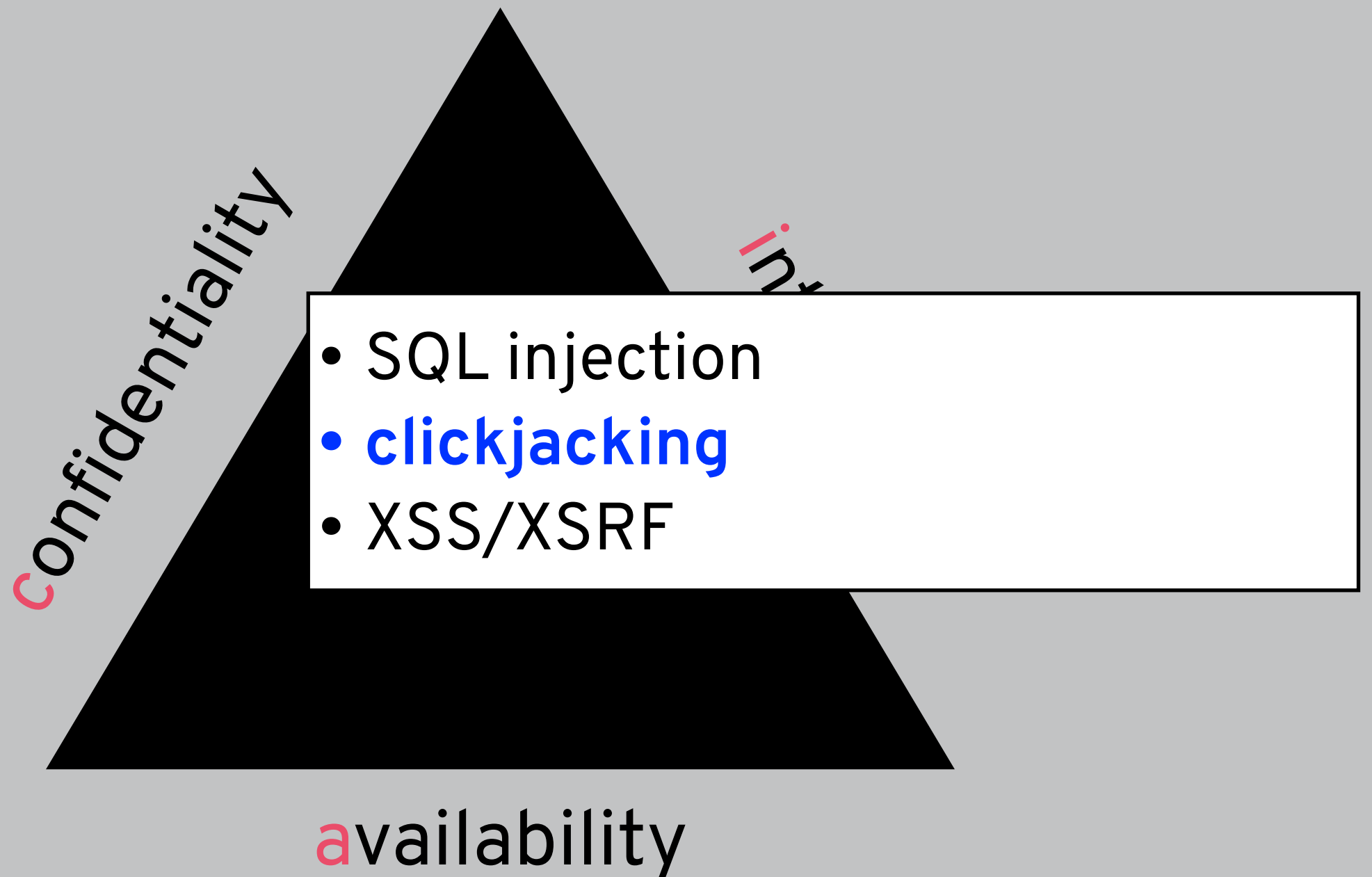
Input sanitisation: make sure only safe input is accepted
What is unsafe?

- Single quote? Dashes? But these could be legitimate

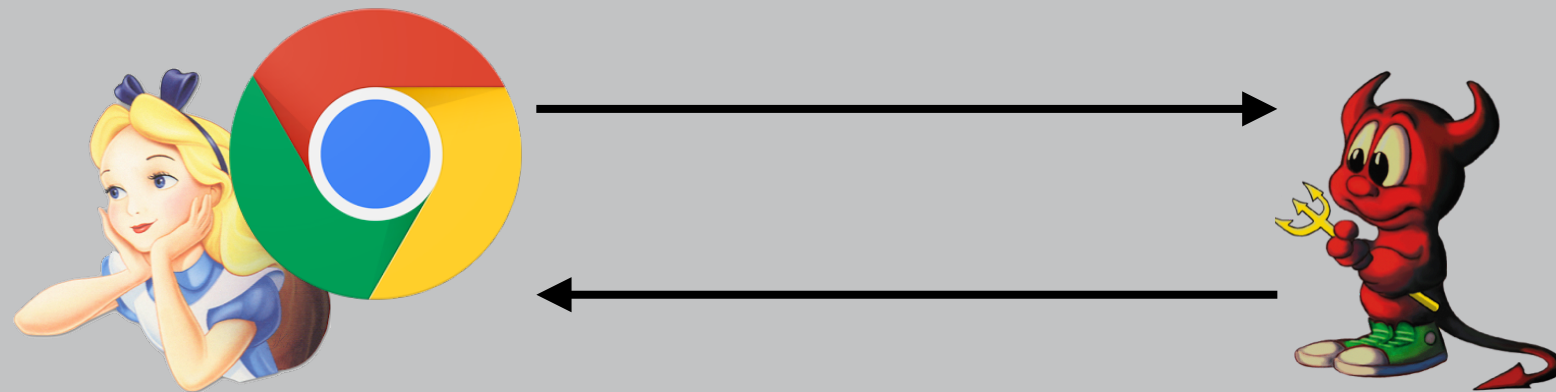
Sanitise on the client or the server?

Use proper escaping/encoding?

INTEGRITY, REVISITED



THREAT MODEL



Is the server trusted by the browser? or the user?

- Browser fingerprinting
- Forward secrecy / revocation
- Typosquatting / pharming
- Clickjacking

HOW DOES THE MODERN WEB WORK?

The screenshot shows the homepage of The New York Times. The masthead features the newspaper's name in a large, black, serif font. Below it, the date "Tuesday, February 27, 2018" is displayed, along with links for "Today's Paper" and "Video". To the right of the date, the weather is shown as "40°F" with a sun icon, and the FTSE 100 index is listed as "+0.02% ↑".

On the left side of the page, there is a red-bordered box containing a photograph of fireworks over a city skyline, with the text "The School of The New York Times" overlaid. On the right side, another red-bordered box contains a similar photograph with the text "Two-week summer programs for high school students.".

Below the masthead, a navigation bar lists various sections: "World", "U.S.", "Politics", "N.Y.", "Business", "Opinion", "Tech", "Science", "Health", "Travel", "Magazine", "T Magazine", and "ALL". The text "these might too" is written in red above the "Travel" and "Magazine" links.

A large, dark gray banner with a red border dominates the lower half of the page. It features a blue icon of a car and a key on the left. The main text in the banner reads "\$8.95/Day Car Rentals" in large white font, followed by "\$8.95/Day Car Rentals Compare & Book Deals Up to 35% Off!" in smaller white font. A yellow button with the text "Book now" is located on the right side of the banner. In the top right corner of the banner, there are small icons for information and a close button.

Three red arrows originate from the text "these come from a different site" at the bottom of the slide. One arrow points to the "U.S." link in the navigation bar, another points to the car rental banner, and the third points to the "Two-week summer programs" box on the right side of the page.

HOW DOES THE MODERN WEB WORK?

The screenshot shows the New York Times homepage with a Ghostery browser extension overlay. Red boxes highlight specific elements: the top left image 'The School of The New York Times', the top right image 'Two-week summer programs for high school students.', the weather and stock information bar, and a large advertisement for '\$8.95/Day'. A yellow text box is overlaid on the advertisement. Red lines connect the bottom text to the advertisement and the navigation menu.

The School of The New York Times

The New York Times

Tuesday, February 27, 2018 | Today's Paper | Video | 40°F | FTSE 100 +0.02% ↑

World U.S. Politics N.Y. Business Opinion Tech Science Health

\$8.95/Day

Everything on `www.nytimes.com` doesn't necessarily come from The New York Times!

these come from a different source

Ghostery

Page Load: 8.62 secs

55 Trackers

TRACKERS

Collapse All

Advertising

Site Analytics

11 TRACKERS 4 BLOCKED

Social Media

2 TRACKERS 1 BLOCKED

Trust Site

Restrict Site

Pause

List View

IFRAMES

Content from one site is embedded into another using **iframes**

Example:

```
<iframe src="https://www.google.com">  
</iframe>
```

framing/outer page

framed/inner page



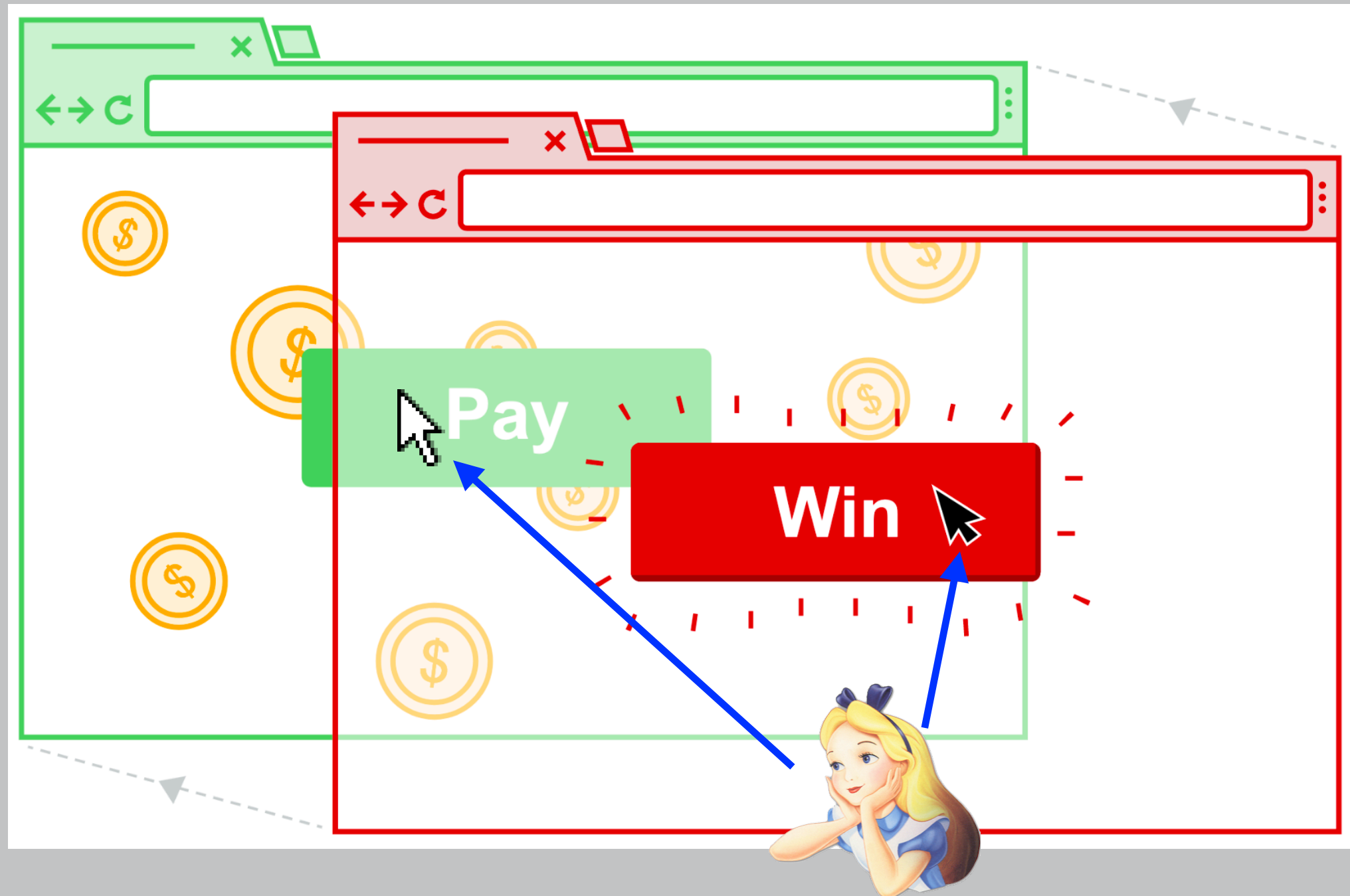
IFRAMES

Content from one site is embedded into another using **iframes**

Outer page can set the width and height of the frame

Only inner page can draw within its frame

CLICKJACKING

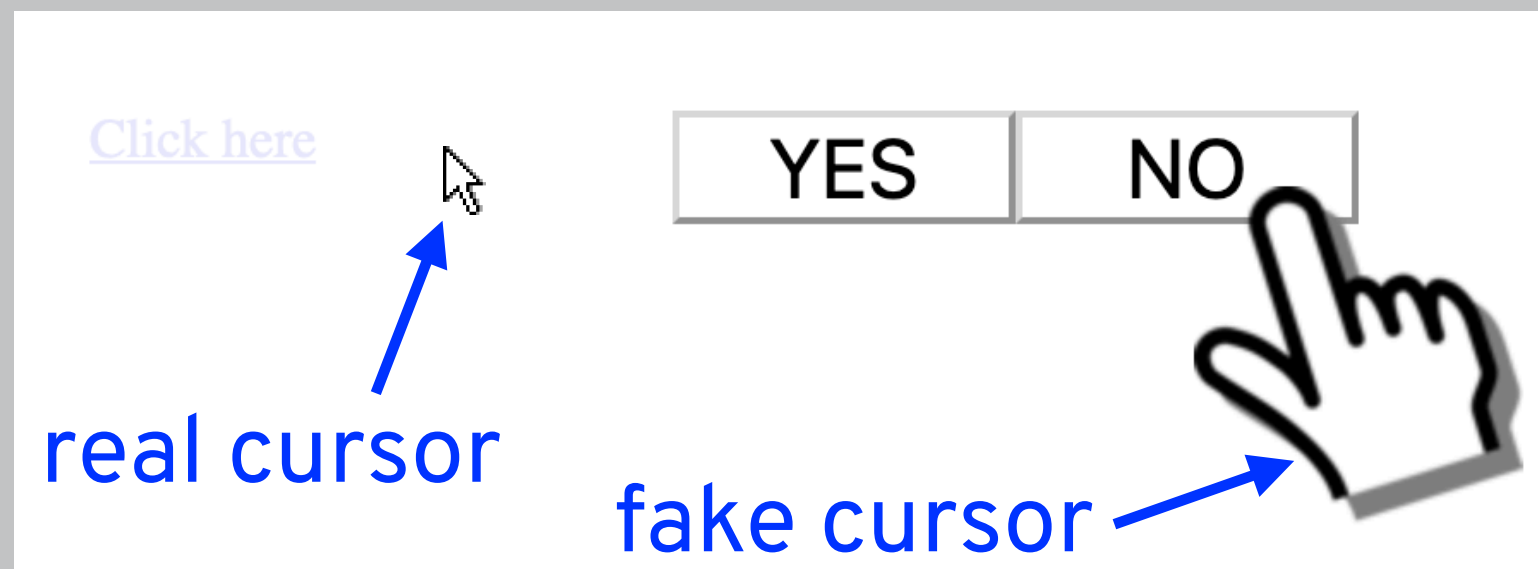


CLICKJACKING

Attacker site frames good site and covers part of it to look different / create unintended interactions

Likejacking is when Facebook users are tricked into clicking the 'like' button

This is often achieved using **cursorjacking**



INTEGRITY, REVISITED

