Data Paging, HTTP Headers and CSRF

Dan Wahlin

WAHLIN CONSULTING

@DanWahlin   www.codewithdan.com

Module Overview

Add a paging header to a RESTful service response

Access headers in an Angular service

Add paging support to a component

Adding a paging component

CSRF Overview

Add CSRF functionality

Using a CSRF token in an Angular service

Adding a Paging Header to a RESTful Service Response

Accessing Headers and Data in an Angular Service

Adding Paging Support to a Component

Adding a Paging Component

CSRF Overview

CSRF

Cross-Site Request Forgery

What is CSRF?

User visits badsite.com in another tab/window (XSS, etc.)

2

3 badsite.com sends a fake page that looks like yourbank.com to the user

4 Victim's browser sends malicious request to yourbank.com with user credentials

1

User logs in to yourbank.com and creates session
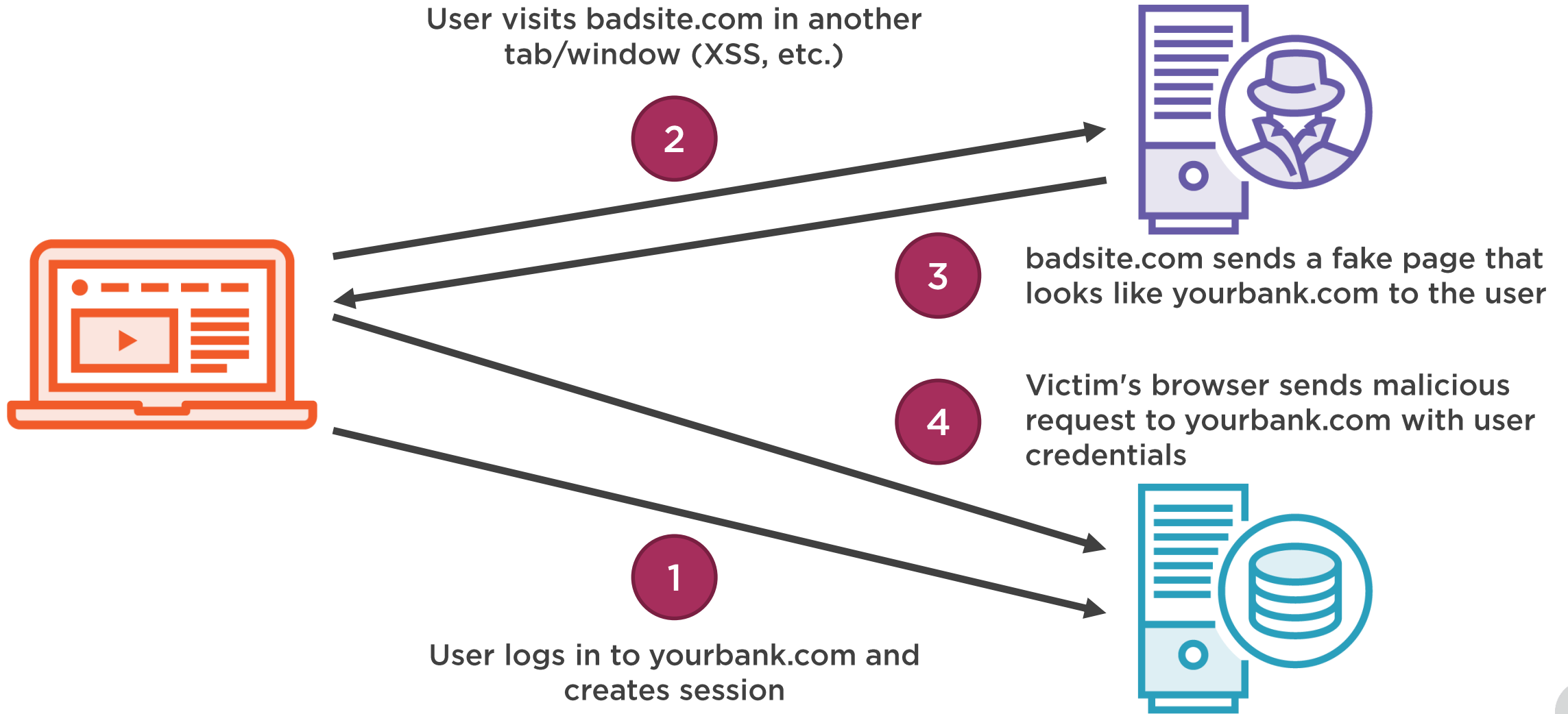
How CSRF Works

Form Post from badsite.com

```
<form action="https://yourbank.com/someOperation" method="POST">
    <button type="submit">Click here for free money!</button>
</form>
<script>
  document.forms[0].submit();
</script>
```

Http/Ajax request from badsite.com

```
<script>
  http.post('http://yourbank.com/api/someOperation', data);
</script>
```

Adding CSRF Functionality

yourbank.com passes a cookie to the client
that contains a CSRF token value

Client sends the cookie back to yourbank.com
AND sets a header with the cookie value in it

Server compares cookie
value to header value

Using a CSRF Token in an Angular Service

Summary

Headers can be used to send and receive data for paging and more

Components provide a reusable way to add paging functionality

CSRF attacks can be shutdown using a combination of a cookie and a header

Angular provides built-in CSRF functionality