Security and Building for Production

Mark Zamoyta
SOFTWARE DEVELOPER AND EDUCATOR

@markzamoyta

Introduction

Chrome Developer Tools and Security

Security and the eval() Function

Preventing Man-in-the-middle Attacks

Cross-site Scripting (XSS)

Building Your Application for Production

Chrome Developer Tools and Security

Application Data Security

Don't store passwords, secrets or other sensitive information

Don't use global variables

Assume hackers can read your JS code and access all data sent to a browser

Security and the eval() Function

Preventing Man-in-the-middle Attacks


HTML
<script>...</script>

Prevent Man-in-the-middle Attacks

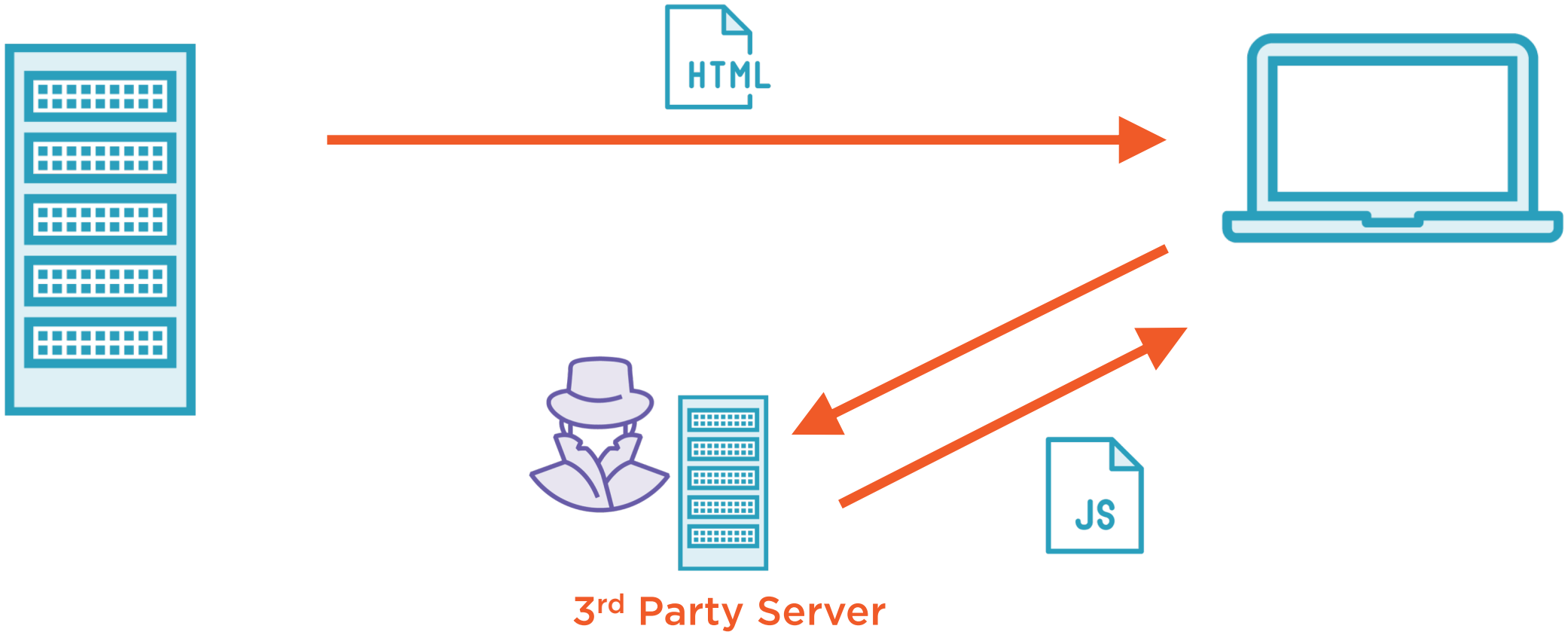
Use SSL

Use HTTP Header: Strict-Transport-Security

Use cookie attributes: Secure and HttpOnly

Cross-site Scripting (XSS)

Cross-site Scripting (XSS)



Addressing Cross-site Scripting Attacks

CSP: Content Security Policy

Use HTTP Header: Content-Security-Policy

CORS: Cross Origin Resource Sharing

Use HTTP Header: Access-Control-Allow-Origin

Building Your Application for Production

Summary

Chrome Developer Tools and Security

- All your code and data is exposed

Avoid JS's eval() Function

Preventing Man-in-the-middle Attacks

- Use SSL related mechanisms

XSS

Building for Production

- npm run build
- /dist folder