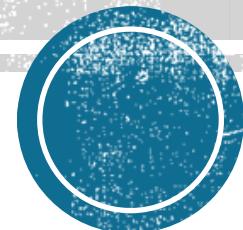
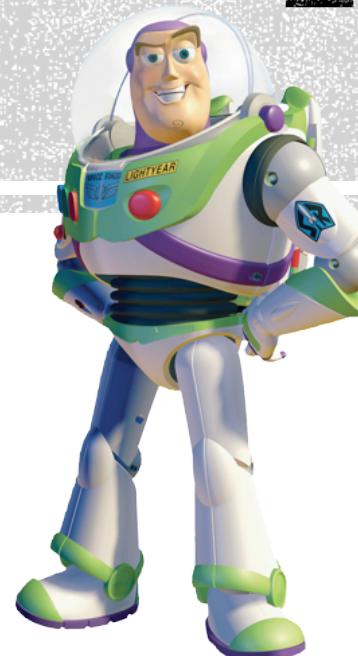


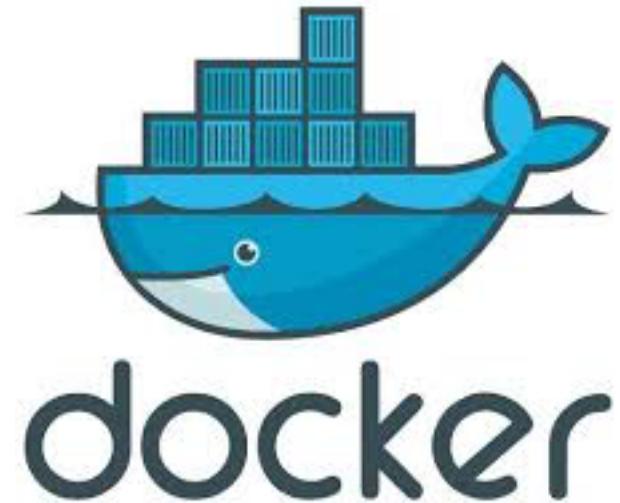
Cross Site Scripting

To alert() and beyond



What?

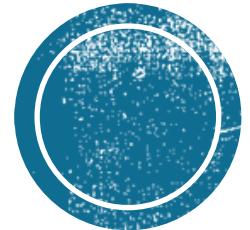
- Introduction to XSS
 - How to find XSS
 - What can be done
 - Tools to help exploit XSS
 - Defense against XSS
- Lots of Demos
 - Play Along - <https://github.com/jbarone/xsslabs>



Nope

- How to code JavaScript
- Single Origin Policy Bypass
- Browser evasion
- Server evasion
- How to hack the Gibson

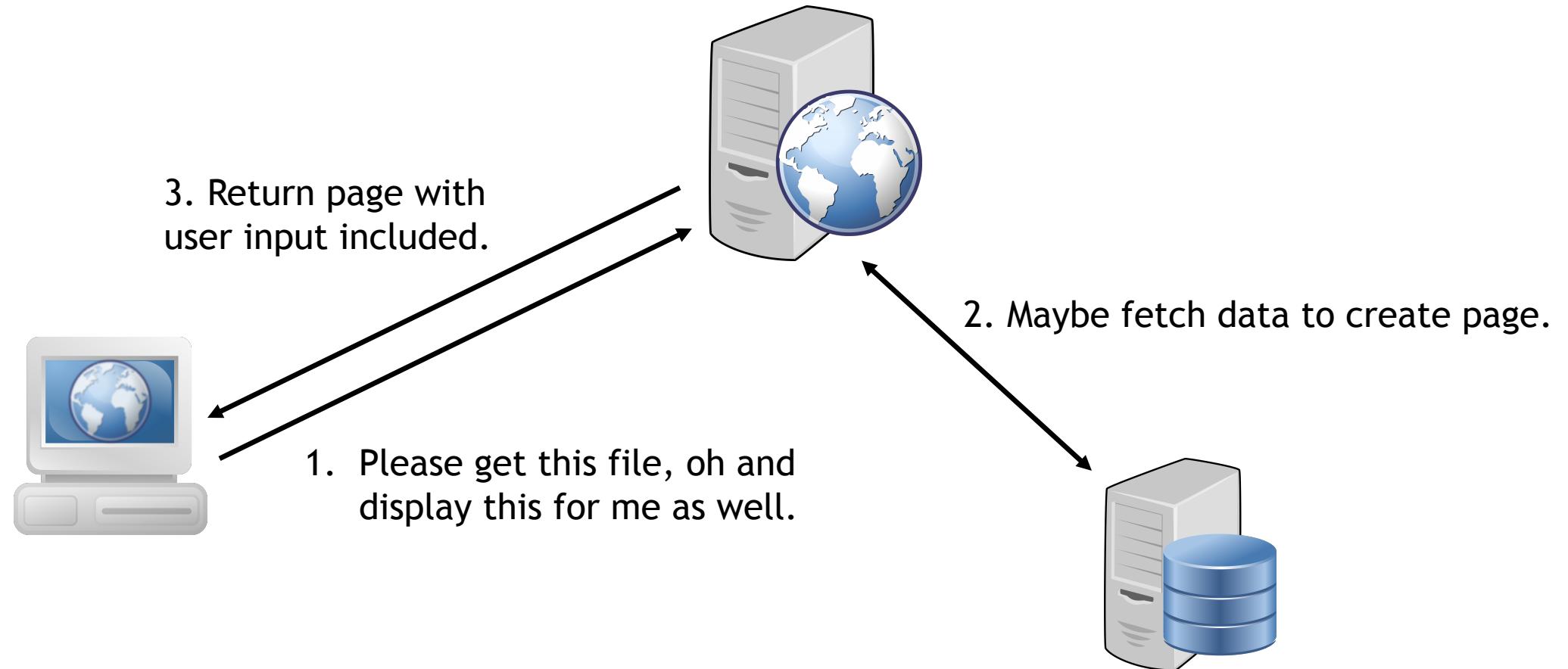




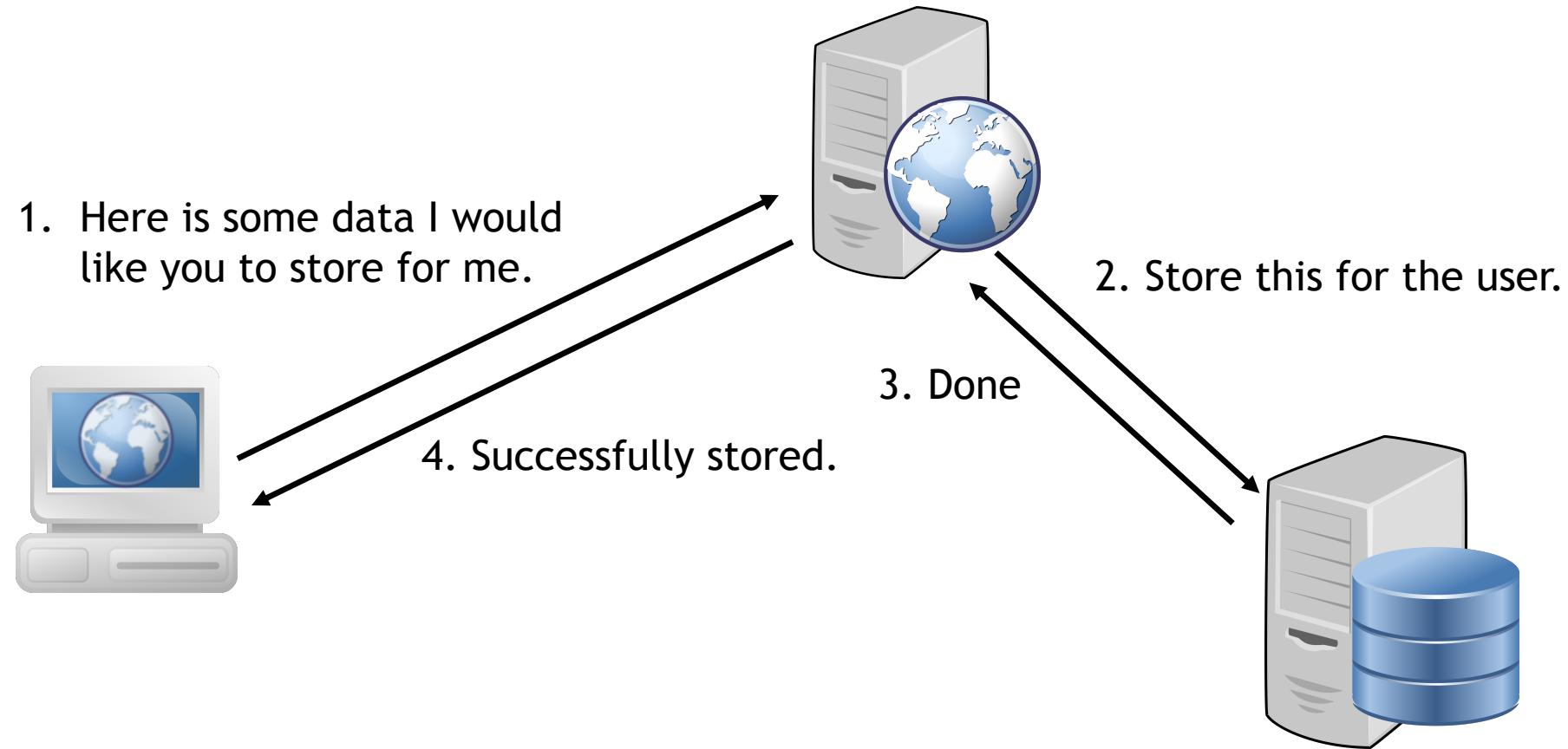
Cross site scripting

Classifications of Attacks

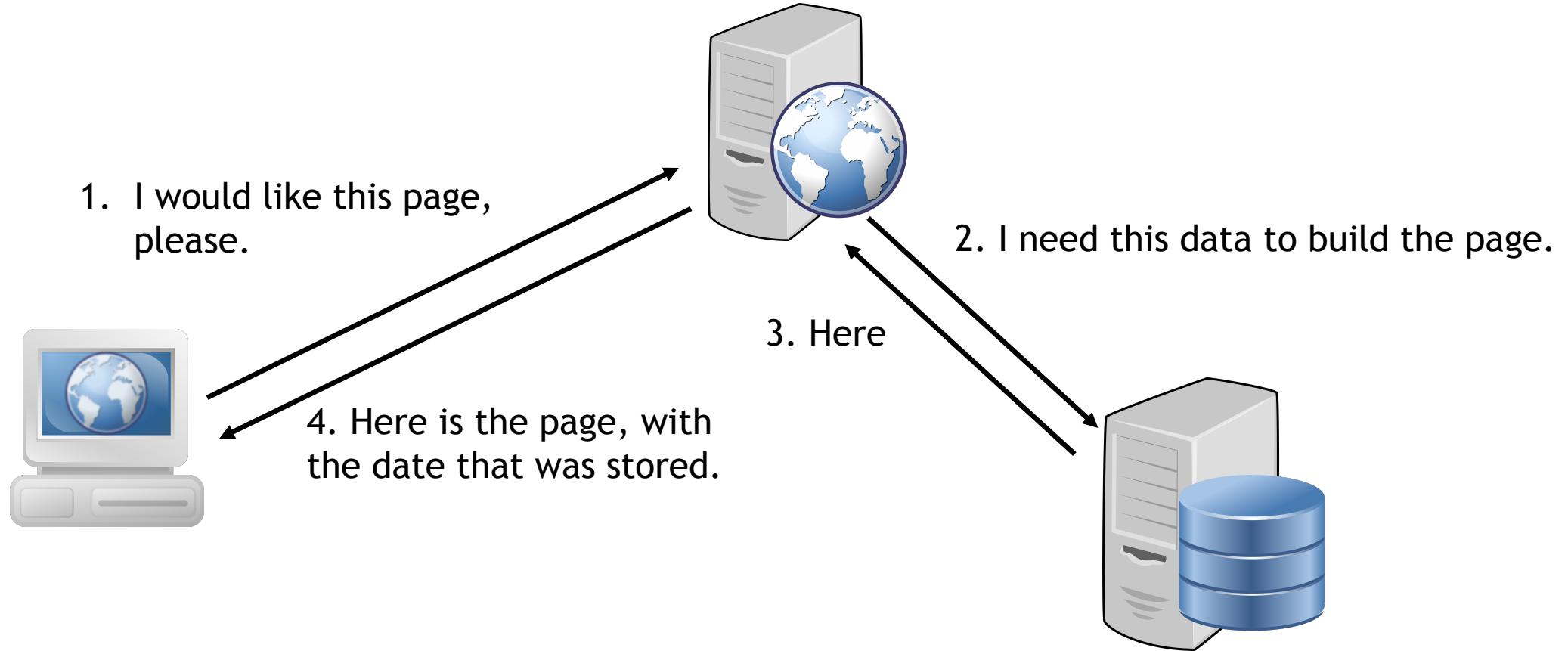
Reflected



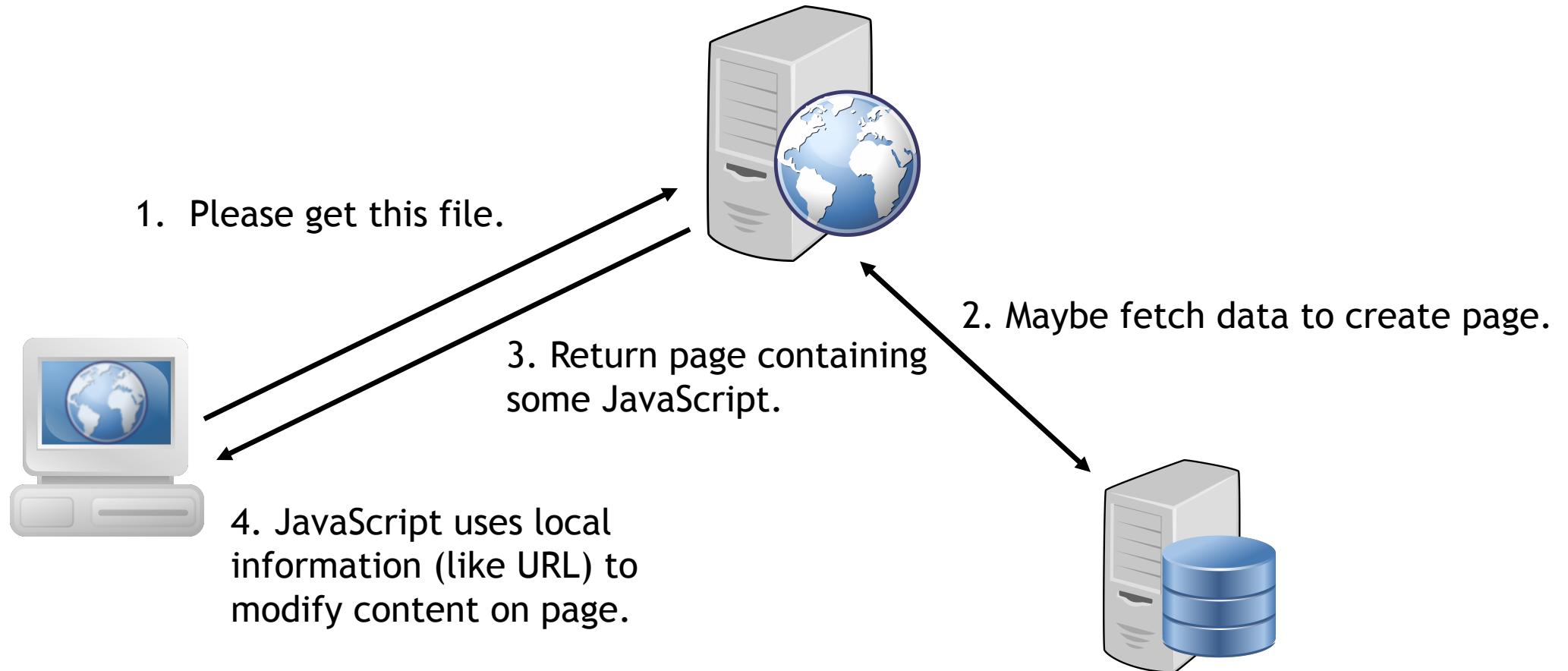
Persistent



Persistent (cont.)

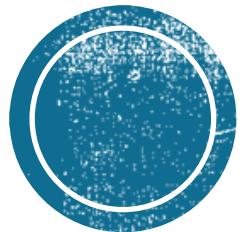


DOM-Based



DEMO TIME!!





Finding XSS



Context matters

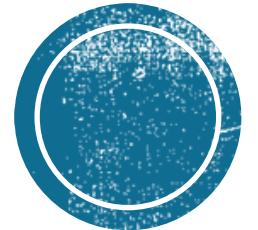
- HTML Injection
- Inside HTML Tag
 - IMG tag src
 - Event attribute tags (onerror)
- Inside JavaScript code block



Tools

- XSSSNIPER - <https://github.com/gbrindisi/xsssniper>
- XSSER (has a GUI) - <https://xsser.03c8.net/>
- XSSCRAPY - <https://github.com/DanMcInerney/xsscrapy>
- Interception Proxies
 - ZAP - https://www.owasp.org/index.php/OWASP_Zed_Attack_Proxy_Project
 - Burp - <https://portswigger.net/burp/>





The Attacks

More than alert()



Attacking Users

- Deface Content
- Capture Input
- Social Engineering
 - Get user to submit passwords
 - Get user to install “software”
- Session stealing
- Session Riding (CSRF)



Attacking Browsers

- Sandbox Escapes
- Heap Exploitation
- Attacks on Extensions
- Attacks on Plugins
- Proxy through the browser
- Metasploit attacks



Attacking Networks

- Ping Sweeps
- Port Scanning
- Fingerprinting services
- Attacking services



DEMO TIME!!



BeEF

- Browser Exploitation Framework
- Designed to exploit XSS
- Focus on payload delivery
- Written in Ruby
- Integrates with Metasploit



BeEF (Continued)

The screenshot shows the BeEF 0.4.7.0-alpha web interface running at 127.0.0.1. The left sidebar lists "Hooked Browsers" under "Online Browsers" (127.0.0.1) and "Offline Browsers". The main area has tabs for "Getting Started", "Logs", and "Current Browser". The "Commands" tab is selected, showing the "Module Tree" on the left with categories like Browser, Chrome Extensions, Debug, Exploits, Host, IPEC, Metasploit, Misc, Network, Persistence, Phonegap, and Social Engineering. The "Module Results History" table shows one entry: id 0, date 2017-04-01 07:06, label command 1. On the right, the "Pretty Theft" module configuration is displayed with fields: Description (Asks the user for their username and password using a floating div.), Id (8), Dialog Type (Facebook), Backing (Grey), and Custom Logo (Generic only) set to http://0.0.0.0:3000/ui/media/images/beef_clippy.png. A "Pretty Theft" logo is also visible above the configuration. At the bottom, there are "Basic" and "Requester" tabs, and a "Ready" status indicator.

127.0.0.1

BeEF 0.4.7.0-alpha | Submit Bug | Logout

Hooked Browsers

Online Browsers

127.0.0.1

Offline Browsers

Getting Started Logs Current Browser

Details Logs Commands Rider XssRays Ipec Network WebRTC

Module Tree

Module Results History

| id | date | label |
|----|------------------|-----------|
| 0 | 2017-04-01 07:06 | command 1 |

Pretty Theft

Description: Asks the user for their username and password using a floating div.

Id: 8

Dialog Type: Facebook

Backing: Grey

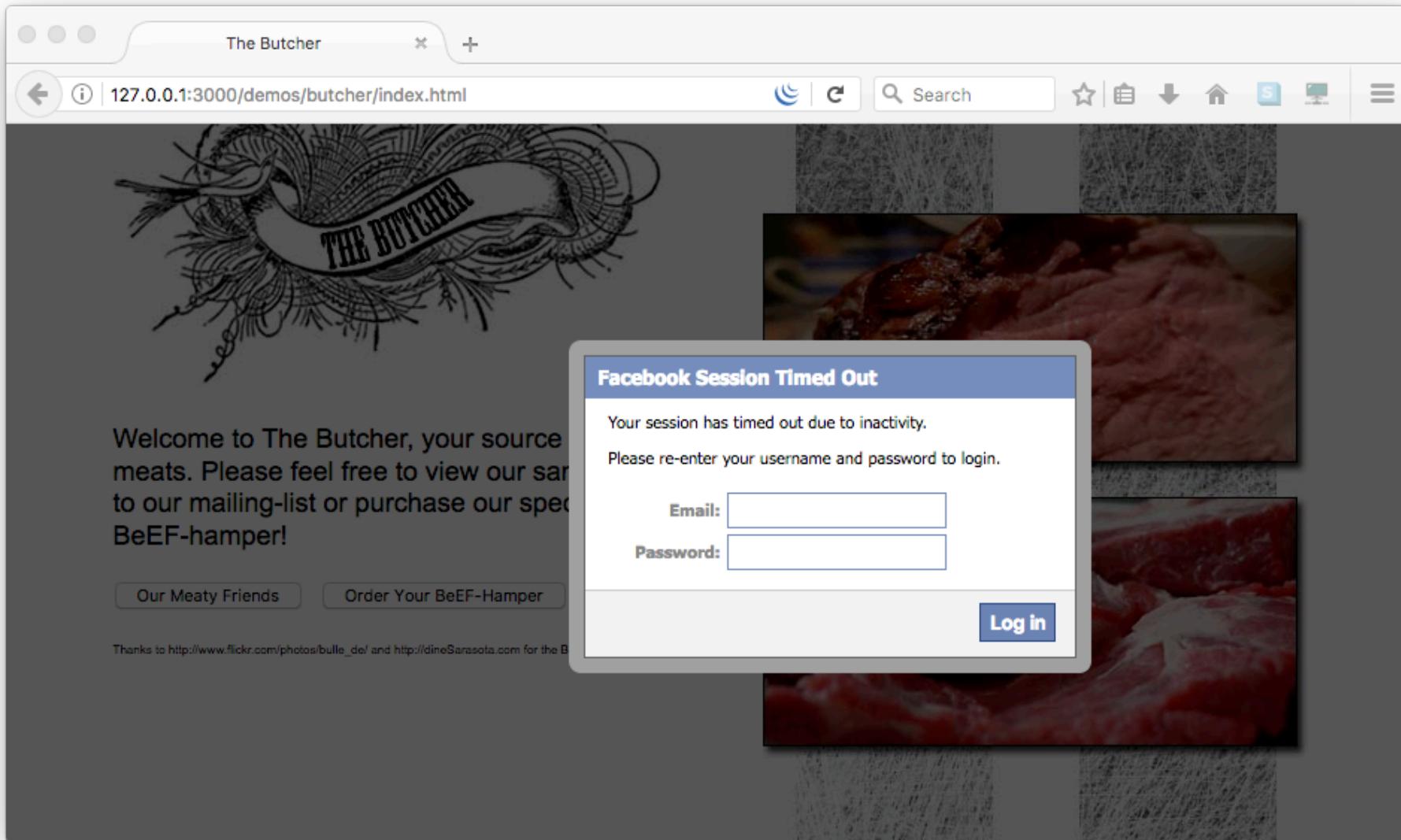
Custom Logo (Generic only): http://0.0.0.0:3000/ui/media/images/beef_clippy.png

Execute

Basic Requester

Ready

BeEF (Continued)





Defense



Defense

- Developers
 - **NEVER TRUST USER INPUT!!!!!!**
 - Sanitize all input (server and client)
 - Sanitize all output
- Users
 - Use browsers with XSS protections
 - Chrome
 - Edge
 - Look before you Click
 - Expand shortened URLs



Thanks

- Joshua Barone
- Twitter: @tygarsai
- Email: joshua.barone@gmail.com
- Blog: <http://caveconfessions.com>
- Lab: <https://github.com/jbarone/xsslab.git>

