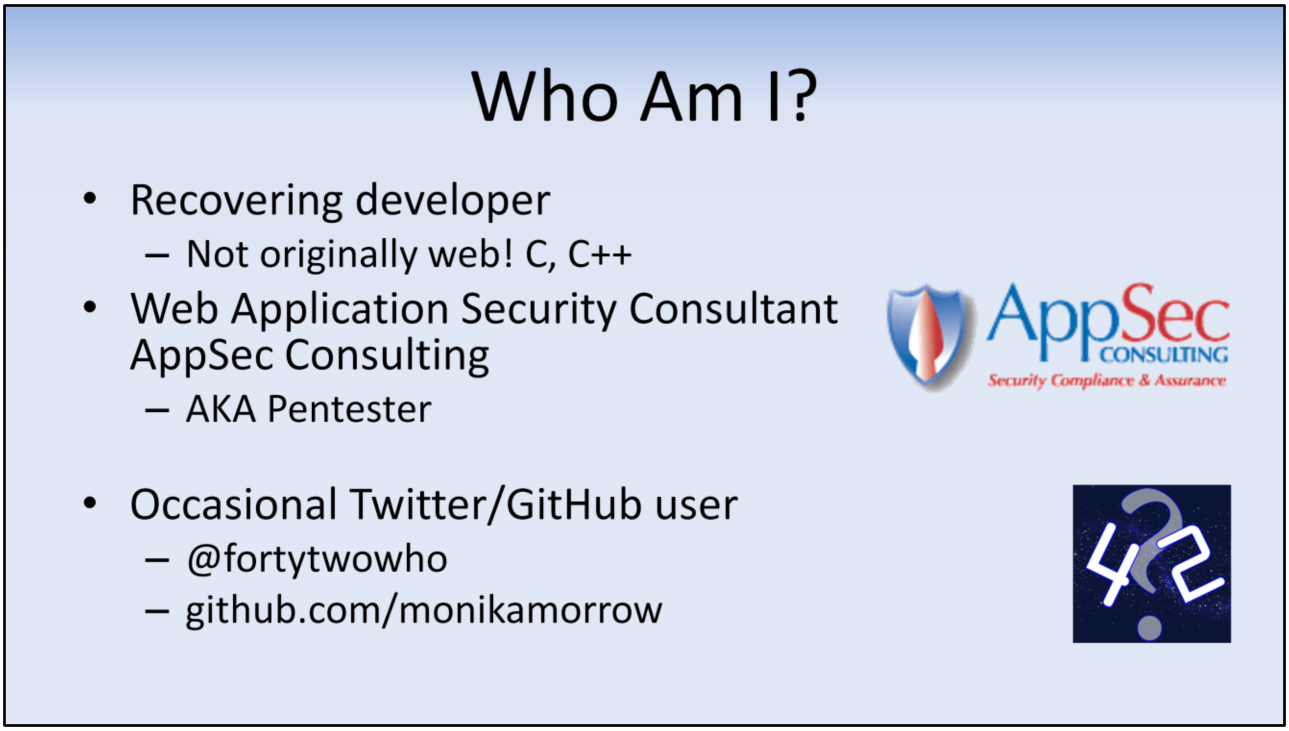
A Peek Behind the Curtain How Pentesters “See” Your Web Application

(and how you can too)



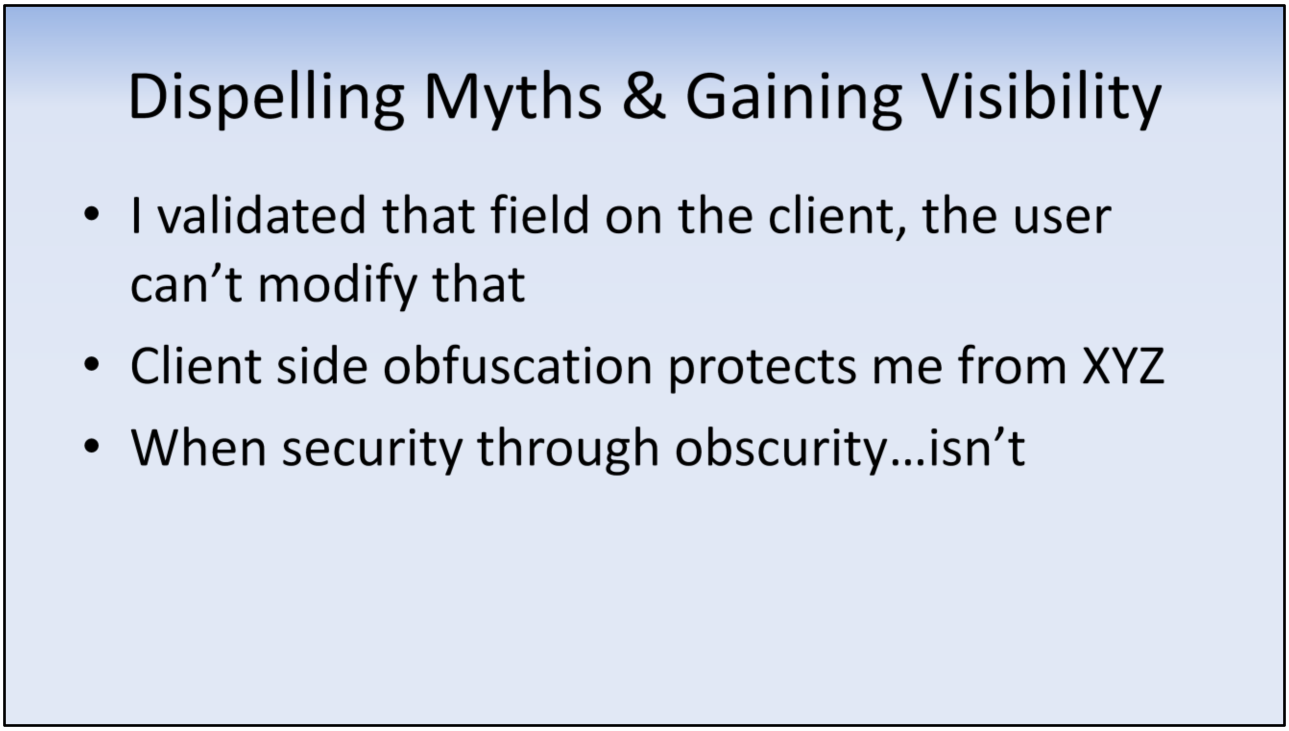
Who Am I?

- Recovering developer
  - Not originally web! C, C++
- Web Application Security Consultant AppSec Consulting
  - AKA Pentester
- Occasional Twitter/GitHub user
  - @fortytwowho
  - github.com/monikamorrow

AppSec CONSULTING
Security Compliance & Assurance
42

Good question! Right up there with the meaning of life. 42! But seriously…

Plaintext links:
https://www.appsecconsulting.com/blog/author/monika-morrow
https://twitter.com/fortytwowho
https://github.com/monikamorrow



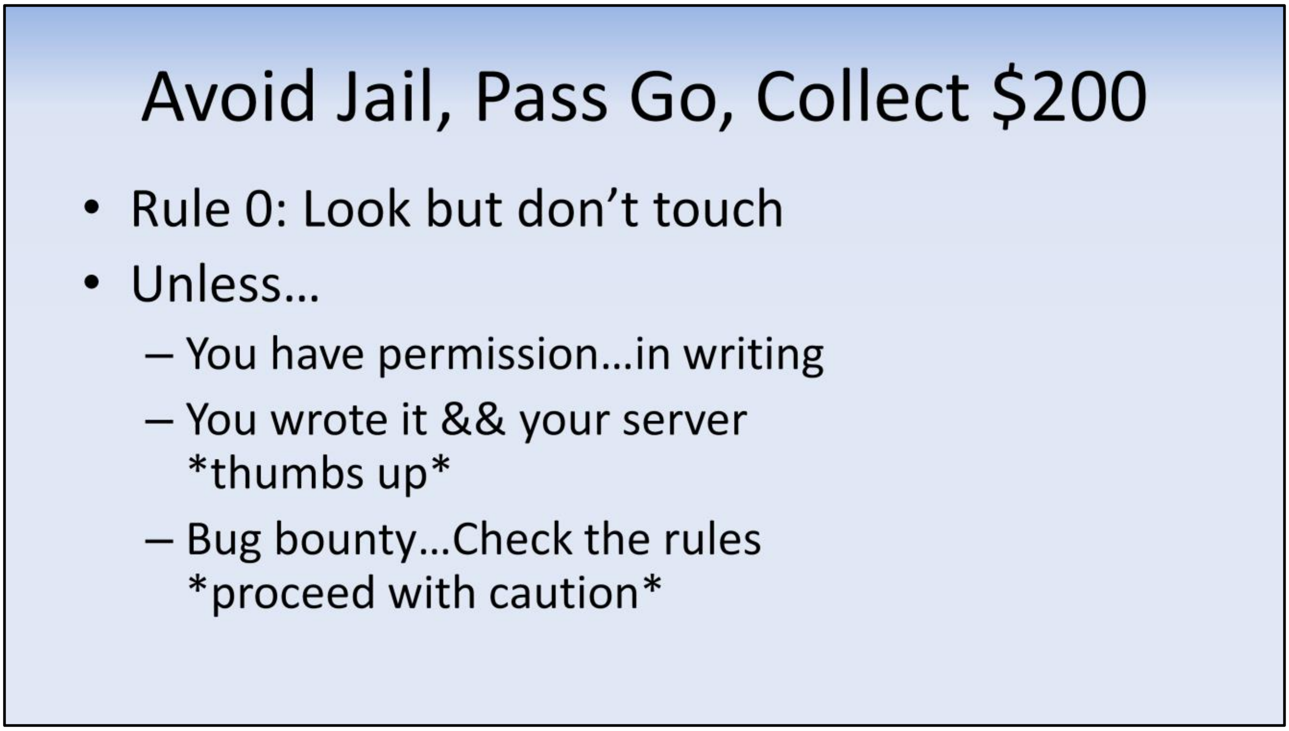
Dispelling Myths & Gaining Visibility

- I validated that field on the client, the user can’t modify that
- Client side obfuscation protects me from XYZ
- When security through obscurity…isn’t

Using Burp we will see that POST parameters can be modified after validation or the proxy can remove validation entirely.

Burp is extensible. Any client side code that is written to obfuscate the communication to the server can be leveraged to have Burp automatically perform the same obfuscation.

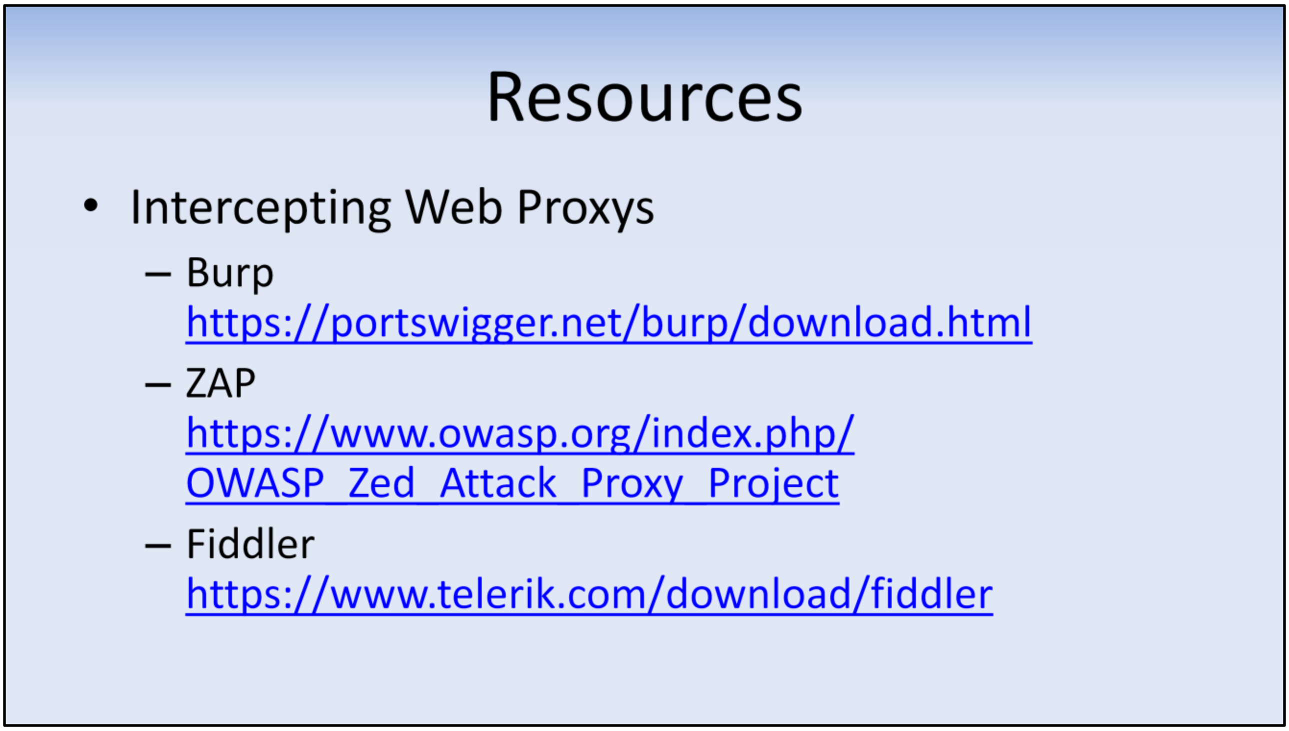
We will see how “hidden” parameters stand out when Burp is configured to modify responses to highlight hidden fields.



Avoid Jail, Pass Go, Collect $200

- Rule 0: Look but don’t touch
- Unless…
  - You have permission…in writing
  - You wrote it && your server *thumbs up*
  - Bug bounty…Check the rules *proceed with caution*

With great power….

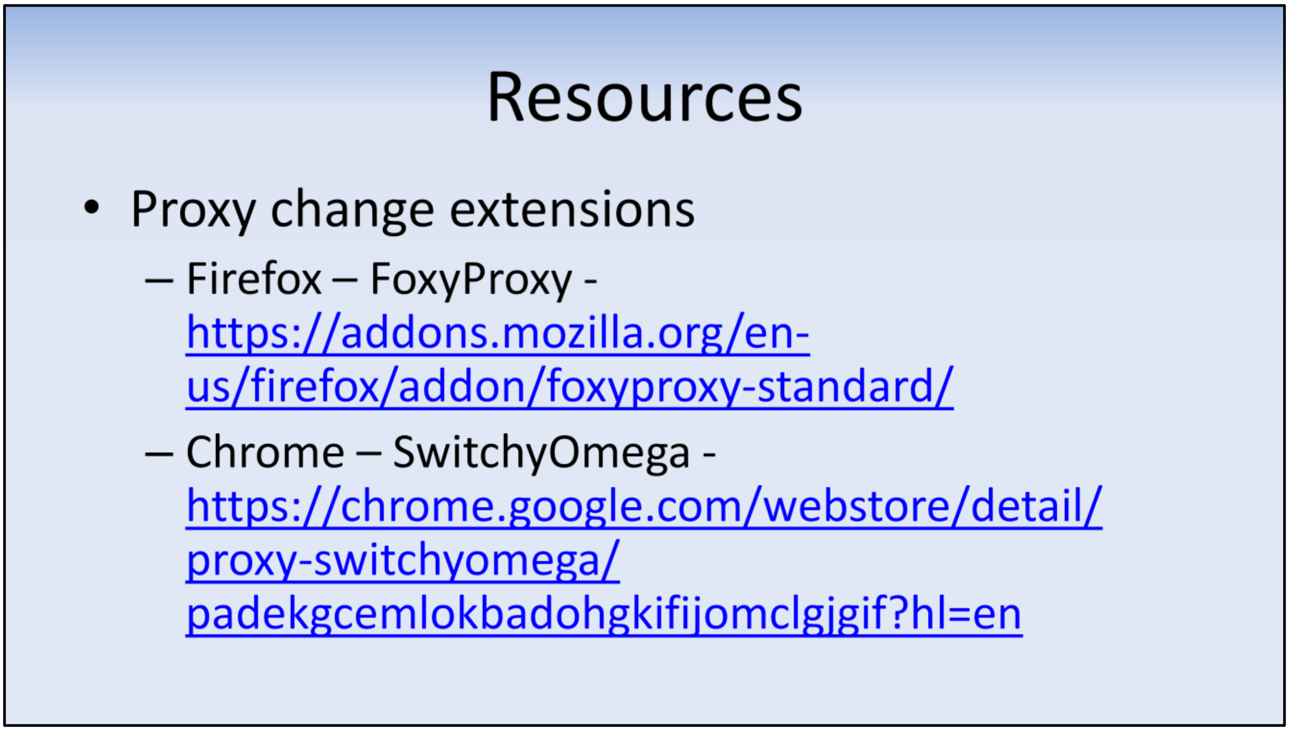


Resources

- Intercepting Web Proxys
  - Burp https://portswigger.net/burp/download.html
  - ZAP https://www.owasp.org/index.php/OWASP_Zed_Attack_Proxy_Project
  - Fiddler https://www.telerik.com/download/fiddler

An intercepting web proxy allows you to view and modify all HTTP(S) traffic between a client and server by maintaining a man-in-the-middle position. The client makes a connection to the proxy (securely if required) and in turn the proxy makes a connection (securely if required) to the server.

Link plaintext:
https://portswigger.net/burp/download.html
https://www.owasp.org/index.php/OWASP_Zed_Attack_Proxy_Project
https://www.telerik.com/download/fiddler



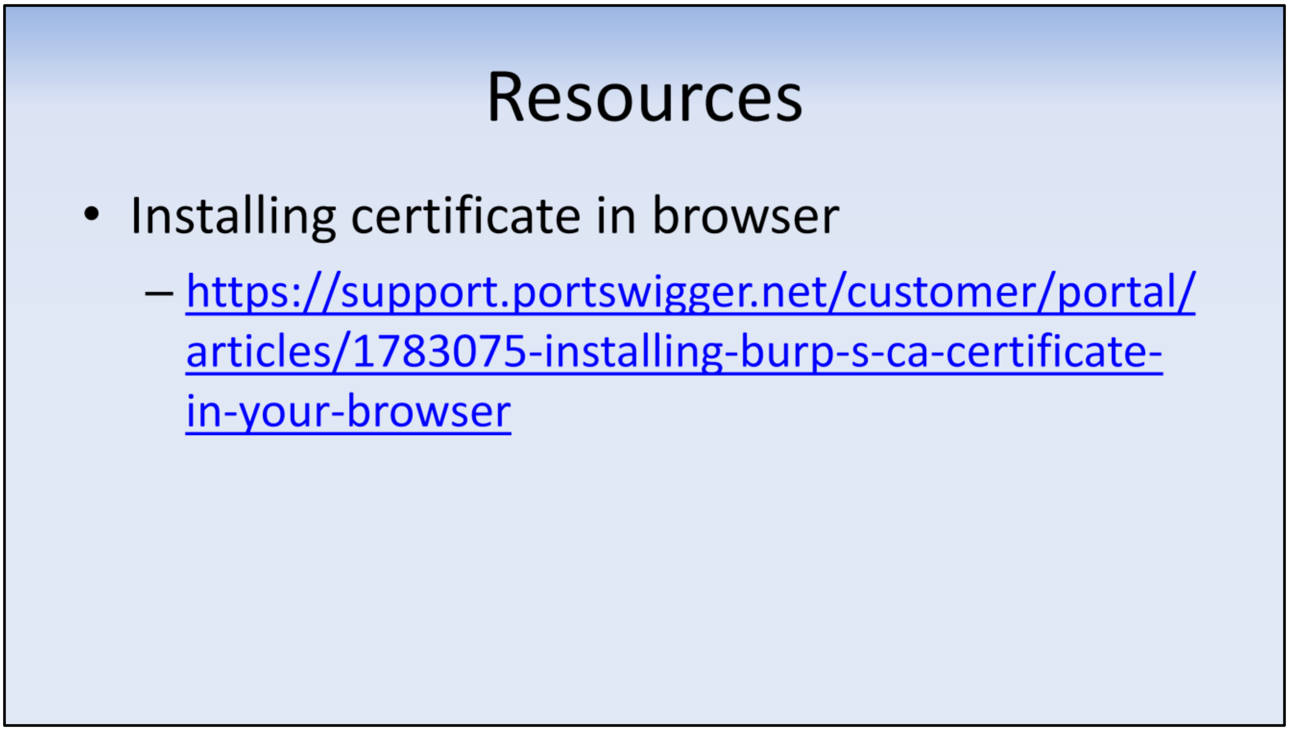
Resources

- Proxy change extensions
  - Firefox – FoxyProxy - https://addons.mozilla.org/en-us/firefox/addon/foxyproxy-standard/
  - Chrome – SwitchyOmega - https://chrome.google.com/webstore/detail/proxy-switchyomega/padekgcemlokbadohgkifijomclgjgif?hl=en

Firefox has its own proxy so it sends only browser traffic to the configured proxy. FoxyProxy makes it easier/faster to switch between configured proxy or no proxy.

Chrome by default uses the system proxy so configuring Chrome to use a proxy sends all system HTTP(S)/Websocket traffic through the proxy. This can make it hard to isolate the traffic you want to examine. The SwitchyOmega extension for Chrome allows you to send only Chrome traffic to your web interception proxy and makes it easy to select a proxy or no proxy.

Link plaintext:
https://addons.mozilla.org/en-us/firefox/addon/foxyproxy-standard/
https://chrome.google.com/webstore/detail/proxy-switchyomega/padekgcemlokbadohgkifijomclgjgif?hl=en



Resources

- Installing certificate in browser
  - https://support.portswigger.net/customer/portal/articles/1783075-installing-burp-s-ca-certificate-in-your-browser

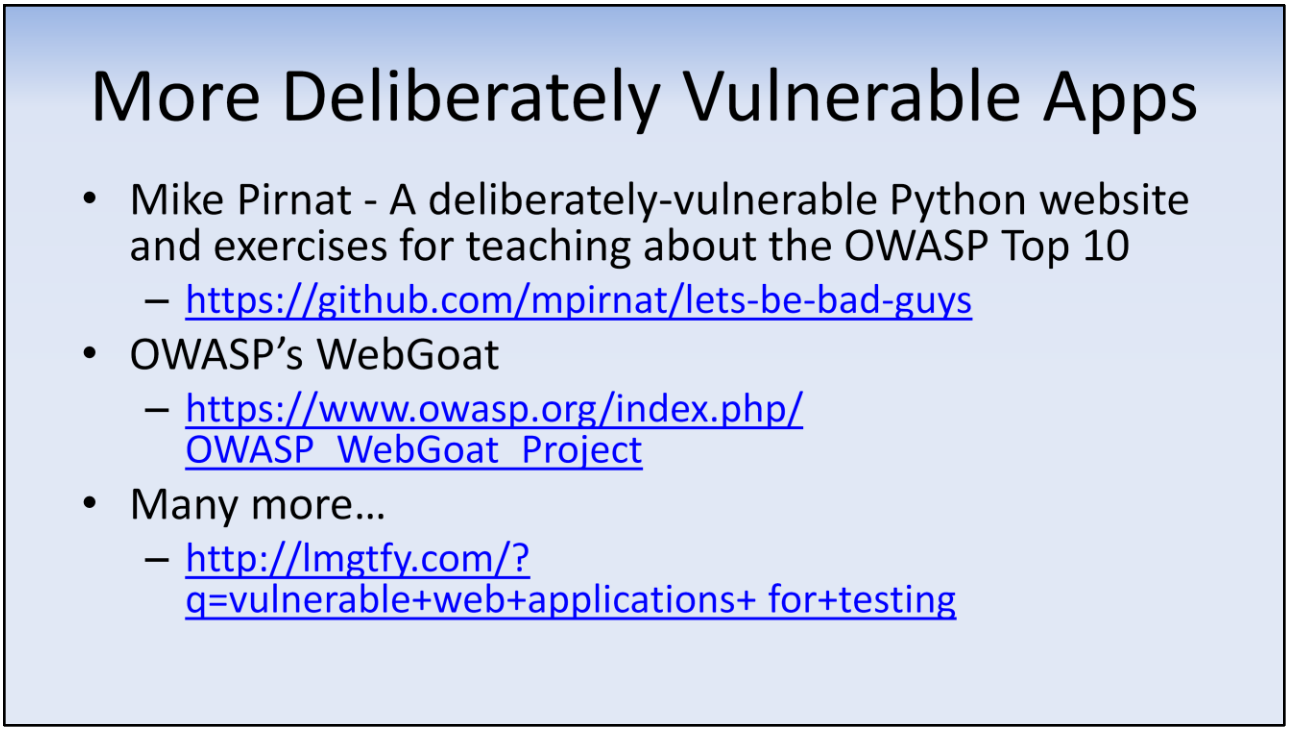
Each installation of Burp has a unique certificate so by installing your certificate in your browser only you will be able to man-in-the-middle your traffic without an obvious TLS warning. Regardless be mindful of when you're passing traffic through the proxy. Using a proxy designed to connect to the broadest range of servers leaves you vulnerable to TLS downgrade attacks that modern browsers have mitigated. Be doubly mindful if you're saving your state. You don't really want to save your plaintext banking password together with your testing data do you?

Link plaintext:
https://support.portswigger.net/customer/portal/articles/1783075-installing-burp-s-ca-certificate-in-your-browser

Okay, Lets Do This Already

DEMO

Troy Hunt – Hack Yourself First
http://hackyourselffirst.troyhunt.com/



More Deliberately Vulnerable Apps

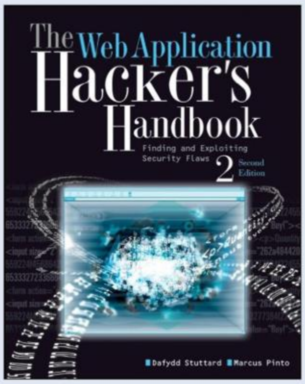
- Mike Pirnat - A deliberately-vulnerable Python website and exercises for teaching about the OWASP Top 10
  - https://github.com/mpirnat/lets-be-bad-guys
- OWASP’s WebGoat
  - https://www.owasp.org/index.php/OWASP_WebGoat_Project
- Many more…
  - http://lmgtfy.com/?q=vulnerable+web+applications+ for+testing

Mike spoke at CodeMash yesterday (Thursday Jan 7, 2016), “Using Python to Get Out The Vote”
https://speakerdeck.com/mpirnat/using-python-to-get-out-the-vote
Also see:
https://speakerdeck.com/mpirnat/shiny-lets-be-bad-guys-exploiting-and-mitigating-the-top-10-web-app-vulnerabilities-2015-edition

Link plaintext:
https://github.com/mpirnat/lets-be-bad-guys
https://www.owasp.org/index.php/OWASP_WebGoat_Project
http://lmgtfy.com/?q=vulnerable+web+applications+for+testing

Want to Learn More?

- The Web Application Hackers Handbook
  http://mdsec.net/wahh/



Link plaintext:
http://mdsec.net/wahh/