Pentesting Web Frameworks
(preview of next year's SEC642 update)

Justin Searle
Managing Partner – UtiliSec
Certified Instructor – SANS Institute
justin@utilisec.com // @meeas

What Are Web Frameworks

- Frameworks for web applications that provide:
  - boilerplate code and templates
  - libraries and tooling
  - session management and backend integration
- Too many web frameworks to count with extensive overlap
- Sub category of web frameworks can be confusing to pentesters
  - Javascript frameworks
  - Frontend web frameworks
  - Content Management Systems (CMS)
  - Backend web frameworks

Why Pentesters Should Care

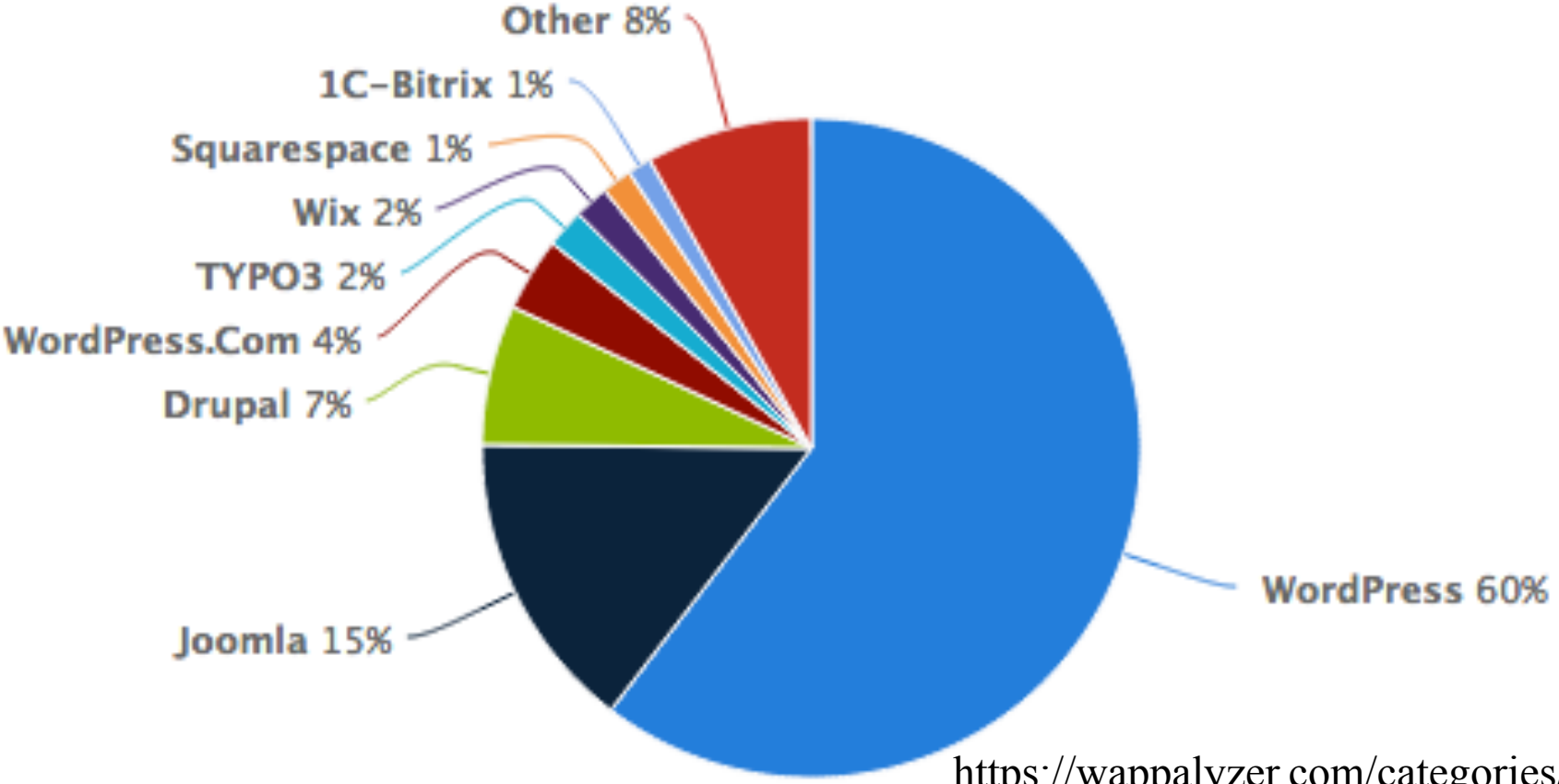
- Web frameworks effects app complexity
  - Increases number of requests and parameters
  - Adds additional layers of obscurity and abstraction
- Familiarity of a framework facilitate pentest optimization
  - Identify vulnerabilities in the framework
  - Test for defaults and known weaknesses
  - Focus testing on custom components

Frontend Web Frameworks

- Frontend frameworks focus on client-side code
- Javascript (AJAX) frameworks
  - JQuery
  - Backbone.js
  - Angular.js
- Frontend (CSS) web frameworks
  - Twitter Bootstrap
  - Zerb Foundation
- Overlap with javascript and frontend frameworks
  - Many Frontend frameworks include JQuery and others
  - Backbone and Angular considered frontend frameworks
- Many push too much business logic to the client

Content Management Systems

- CMS are more of an application category than framework
- Sharepoint is the most popular internal CMS, externally:



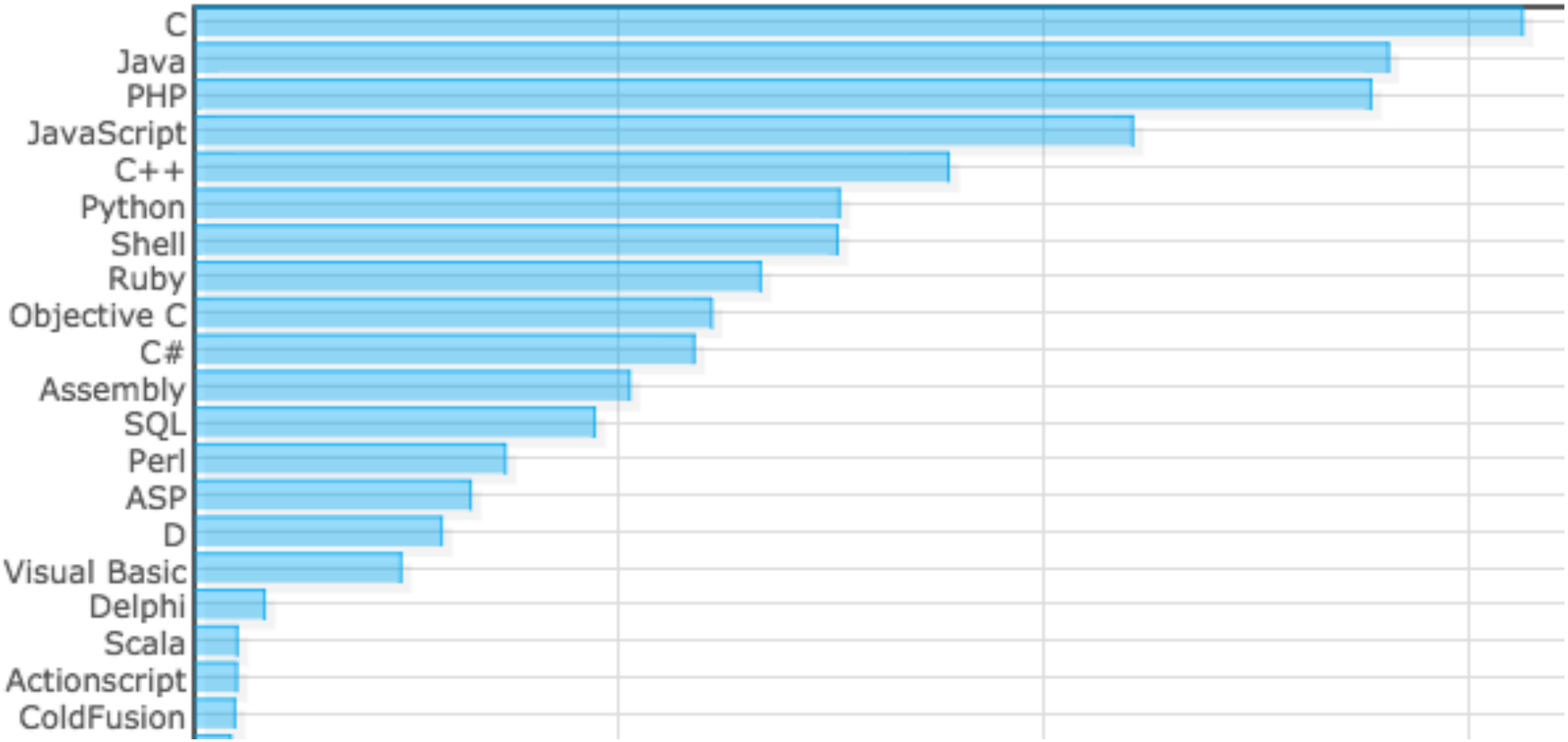
https://wappalyzer.com/categories/cms

Backend Frameworks

- Backend frameworks focus on server-side code
- .NET Framework
  - Active Server Pages (ASP.NET), ASP.NET MVC, MonoRail
- Java
  - Struts, Spring MVC, Java Server Faces (JSF), Google Web Toolkit (GWT)
- PHP
  - CakePHP, Zend Framework, Laravel, CodeIngniter, Symfony
- Python
  - Django, Flask, Pyramid, Zope, Pylons, CherryPy
- Ruby
  - Rails, Rack, Synatra, Padrino

Popularity Based on Language

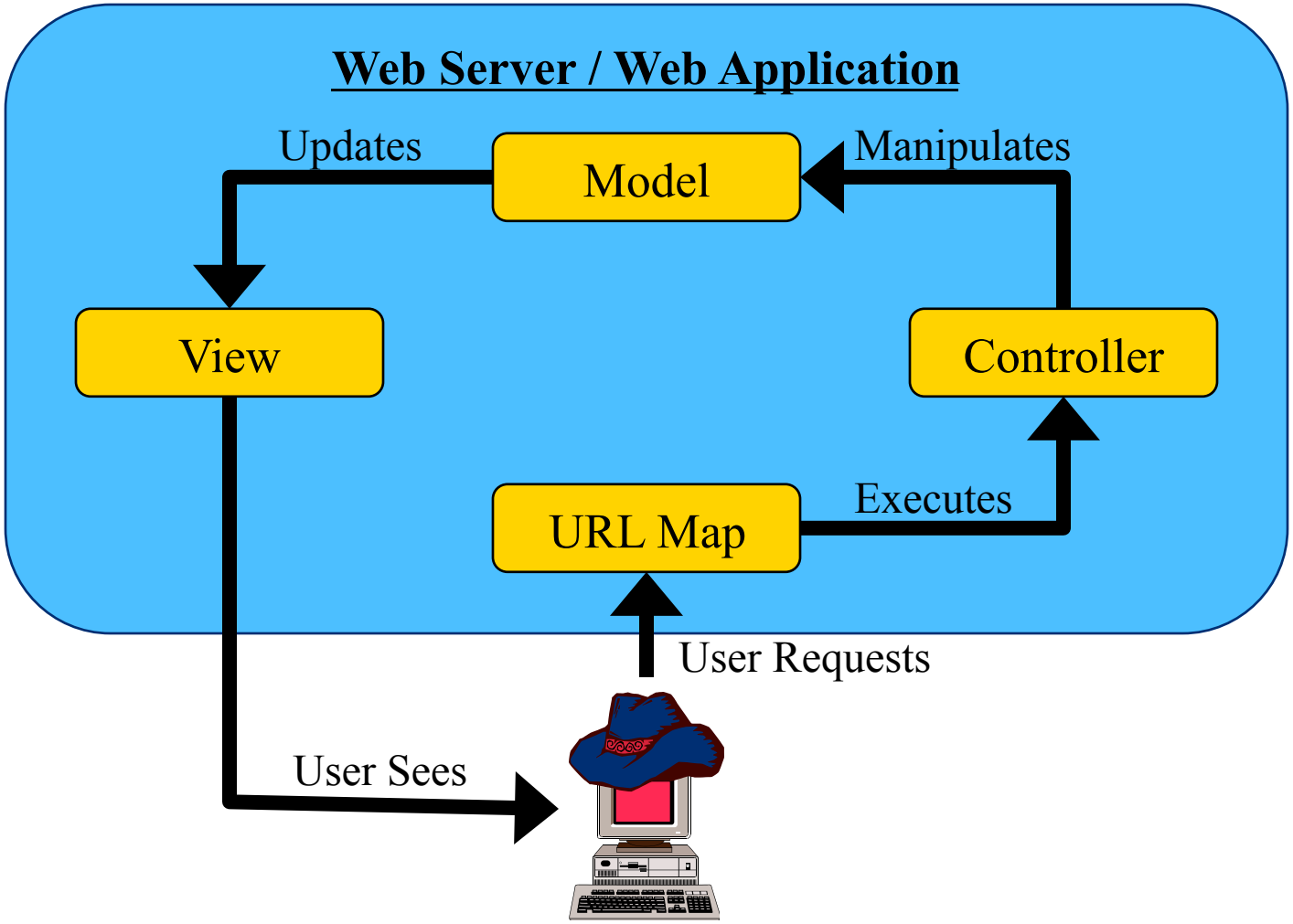
- Not specific to web application development
- http://langpop.com



What Backend Frameworks Provide

- Most use a Model-Viewer-Controller (MVC) architecture
  - Separates application into three functional components
  - Often includes URL mapping (also called routing)
  - Many new frontend frameworks have migrated to MVC on the client
- Model
  - Maintains data objects and application state
  - Can be done in memory, database, or other data store
- Viewer
  - Handles look and feel of the site (desktop vs mobile version)
  - Often uses a template engine or use another frontend framework
- Controller
  - Handles the inputs and provides core business logic

MVC Process Flow



Old School Java (15 years ago)

- Servlet:
  - Java code with http API for input and output
- JSP (java server page):
  - HTML template compiled into a servlet
  - http://example.com/checkout.jsp
- Common pitfalls (100% custom code):
  - XSS: no functions to validate or output encode
  - SQLi: all databases requests hand written
  - All other common web vulns (depends on dev skills)

Struts 1

- Usually uses .do resource extensions
- Model View Controller(MVC) Framework and an "Action-based framework"
  - Action runs on the input
  - Action stores data in the "model"
  - Action returns one named result
  - Different results are wired to different views
  - View usually uses jsp or freemarker
  - urls are not 1-to-1 with views

Struts 2 changes

- Usually uses .action resource extensions
- User input -> action in addition to a form
  - form in struts2 is called a model
  - developers usually just call the action (easier)
- Object (Object Graph Notation Language)
  - Input parameters are OGNL statements
  - run on the action or model
  - equivalent to java in expressiveness

Struts and Input Validation

- Struts uses the Struts Validation Framework configured in the in WEB-INF folder
  - validator-rules.xml
  - validator.xml
- Apache provides common validation rules
  - commons-validator.jar
  - http://commons.apache.org/validator/
- Secondary defense for many different vulnerabilities
  - Pentesters can retrieve default rules file to evaluate input filters
  - Can use a known filtered string to test which inputs are validated

Struts and Output Encoding

- Some Struts tags include output encoding
- Provided by most text objects like:
  - html:TextArea
  - html:File
  - html:Password
  - html:Hidden
  - html:Radio
- Also provided for common write functions
  - bean:Write
  - html:Message
- Helps to prevent XSS vulnerabilities

Struts Roles

- Roles help provide authorization to objects

```
<action
      roles="administrator,contributor"
      path="/article/Edit"
      parameter="org.article.FindByArticle"
      name="articleForm"
      scope="request">
        <forward
              name="success"
              path="article.jsp"/>
</action>
```

Why Should Pentesters Learn the Frameworks?

- Half of all Struts 2 CVEs come from OGNL
- Vulnerable OGNL Injection Code (CVE-2013-2135):

```
<result type="httpheader">
    <param name="headers.foobar">${message}</param>
</result>
```

- OGNL Injection Exploit Request:

```
http://localhost:8080/example/HelloWorld.action?message=${%{1+2}}
```

- Results in OGNL executing the `%{1+2}` code and writing the number 3 back to the user's browser
- Part of a number of injection vulnerabilities found/fixed in 2013
- One may have been used to exploit Apple Developer site

Contact Information

To register for the next SEC642 course, visit:

https://www.sans.org/course/sec642

Justin Searle
personal: justin@meeas.com
work: justin@utilisec.com
cell: 801-784-2052
twitter: @meeas