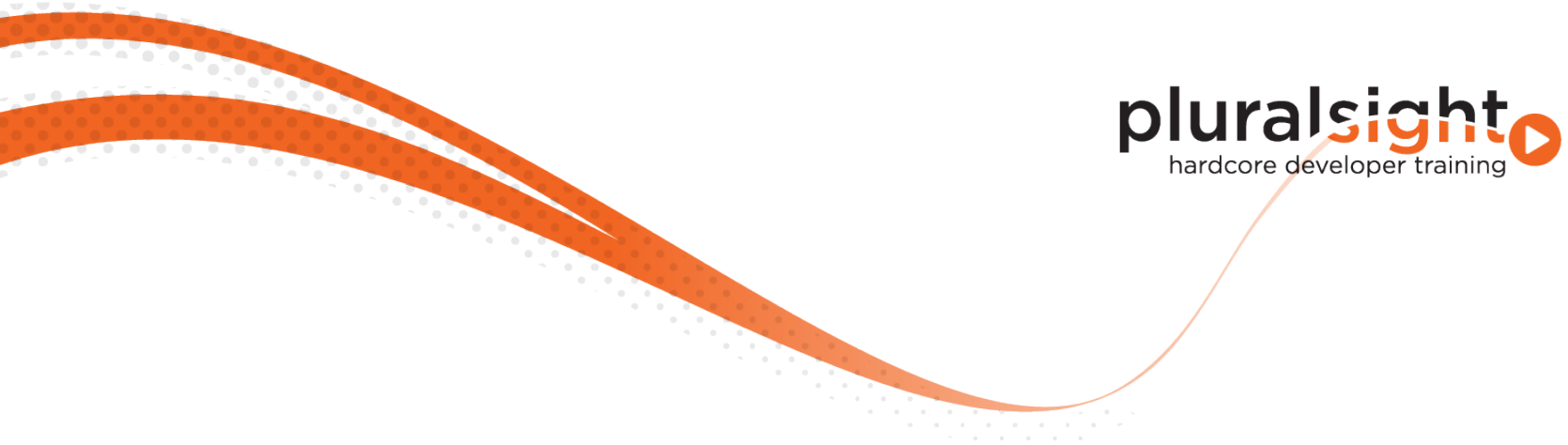
Failure to Restrict URL Access

Troy Hunt
troyhunt.com



pluralsight
hardcore developer training

Outline

- How OWASP views the risk
- Performing an attack against a vulnerable application
- Understanding access controls in ASP.NET
- Leveraging role based authorisation with the role provider
- Other access controls risks and misconceptions

OWASP overview and risk rating

Threat Agents
—
Anyone with network access can send your application a request. Could anonymous users access a private page or regular users a privileged page?

Access controls in ASP.NET

- **Within ASP.NET we have a number of ways of implementing access controls at various levels**
  - At a path level using the location element in the web.config
  - At a method level in any class using principal permissions
  - At an entire MVC controller class using the “authorize” attribute
  - At an MVC controller action using the “authorize” attribute
  - At any point in code using the identity and role features of ASP.NET

Resources where access controls are frequently overlooked

- **APIs that are called asynchronously or via mobile devices**
  - The visibility of these is low; it's not usually a URL people see in their browser
  - However, it's very easy to monitor background or async requests made by the browser
  - Often, an APIs website may *never* be seen directly in the browser
- **Resources not usually loaded in the browser are often neglected**
  - Documents such as Word or PDF are prime candidates
  - Data persistence in text of XML files
  - They may be loaded directly from the file system and not be requested through a handler
  - The integrated pipeline in IIS 7 means these can be protected in the same way as other resources in an ASP.NET app

Common access control misconceptions

- **Remember what *is not* an access control:**
  - Obfuscated URLs
  - Websites without a domain (IP address only)
- **Also be wary of URLs with credentials in them**
  - There are multiple points where they may be cached or intercepted
  - This includes HTTPS addresses

Summary

- **There are many ways to implement authorisation in ASP.NET**
  - Web.config locations, authorise attributes on controllers and actions or anywhere else using User.IsInRole
  - Remember to think about whether you're securing the *path* or the *resource*
- **Role based authorisation is extremely simple using the native features built into all ASP.NET sites**
  - It's easy to either configure it in the DB or hook into the role provider within code
- **Remember there are some "gotchas" within access controls**
  - Don't forget to protect APIs and non-ASP.NET resources
  - Consider all publicly facing resources as public if no access controls exist