Securing Angular Apps

Brian Noyes
CTO, Solliance Inc.
www.solliance.net

solliance
your solution experts.
www.solliance.net

solliance
www.solliance.net

About Brian Noyes

- CTO and Co-founder, Solliance
  www.solliance.net
- Microsoft Regional Director
- Microsoft MVP
- Pluralsight author
  www.pluralsight.com
- Web API Insider, Windows Azure Insider, Window Store App Insider, C#/VB Insider

Microsoft Regional Director

pluralsight

MVP
Microsoft Most Valuable Professional

brian.noyes@solliance.net
@briannoyes
http://briannoyes.net

solliance

Thank Our Sponsors without whom Today is not Possible

Platinum

Silver

Bronze

solliance

Housekeeping…

- Thanks to our host
- Respect your speal
  Set mobile devices t
- Fill out session evaluations
  - They are used in the drawings
- You must be present to win at the wrap-up

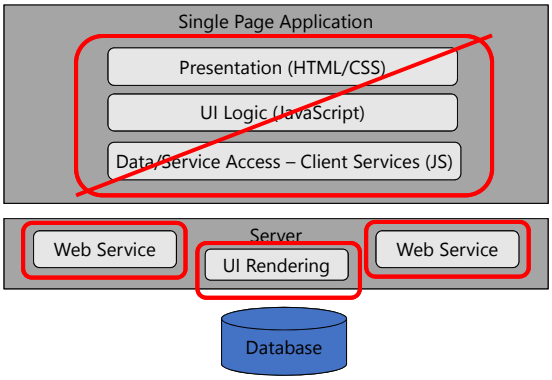
Agenda

- Security in the SPA Architecture
- Protecting against CSRF
- Dealing with CORS
- Client security context

What does it mean to “secure”?

- More than just “logging in”
- Authentication
- Authorization
- Transport protection
- Cross Origin Resource Sharing (CORS)
- Cross Site Request Forgery (CSRF/XSRF)
- Cross Site Scripting (XSS)
- User and access control management

Single Page Application Architecture



Authentication Options

- Windows authentication
- Cookie-based authentication with host site
- Basic authentication
- Token-based authentication (STS)

Securing SPA Pages

- Leverage server page rendering security
- Block return of root SPA page
- Block return of HTML fragments and/or JavaScript
- Only really makes sense if the structure or static content of your pages are sensitive
  - Most content in a SPA delivered as “data” via Web API calls

Securing Web API Calls

- Need to decide on authentication mechanism
  - No redirects for login for service calls – must present valid authorization token
  - Cookie or Authorization header
- Set up depends on your back end technology
- Up to server to allow the calls or not
  - Validates the token based on shared secret trust relationship with the Authorization Server
  - Might supplement the Authorization Server claims with more fine grained app specific claims

Protecting against CSRF

- Cross Site Request Forgery
- Important concern with SPAs due to prevalence of Web API calls, potentially to other service hosts
- Serious threat when using cookie authentication with AJAX calls
  - Need second token through separate means to correlate with cookie
- ASP.NET MVC has built in support
- Web APIs require manual means
- OAuth helps to avoid

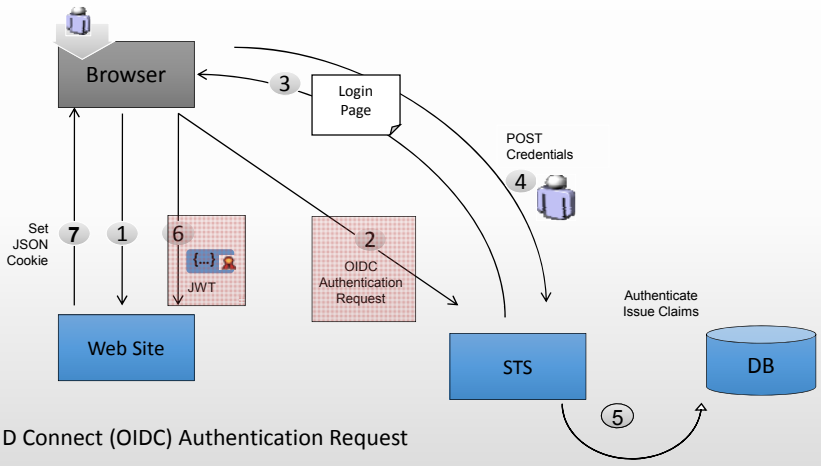
Dealing with CORS

- Cross Origin Resource Sharing
- Web APIs on a different host than pages rendered from
- Built in to all modern browsers
- Simple CORS
  - GET/POST, form encoded, no additional header
  - Sends Origin header in request, expects Access-Control-Allow-Origin in response
- Most CORS
  - Sends “preflight” OPTIONS request specifying what is being requested (Verb, headers, cookies,etc)
- Destination server decides who gets in
- Have to populate appropriate headers in your $http service calls
- Automatic with Angular $http service with right configuration

Token Authentication Options

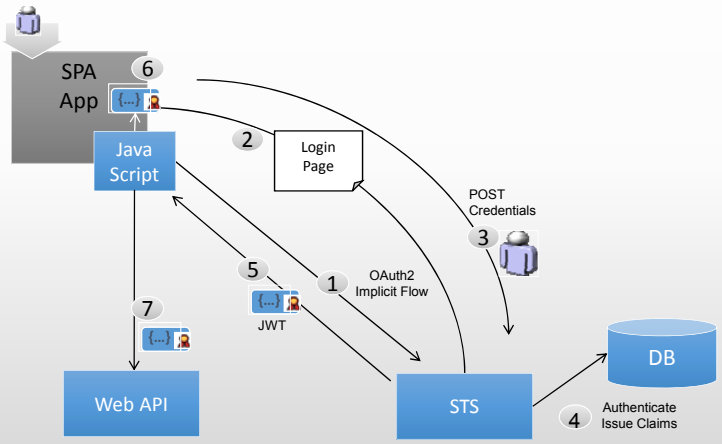
- .NET Backend
  - ASP.NET Identity
  - IdentityServer
- NodeJS Backend
  - Passport, etc.
- Commercial option
  - Auth0

Token-based Redirect



OpenID Connect (OIDC) Authentication Request

OAuth2 Implicit Flow



Client Security Context

- Client may collect credentials to send to authentication server for validation
  - Resource owner password flow
  - Discouraged in OAuth2 spec
- Can track success or failure of login process
- Can obtain claims from returned tokens
- Can request server authorization roles/claims
- Should only be used to drive client UX – not treated as "securing the app"
  - Hide/show navigation links
  - Enable features

Protecting Against XSS

- ngSanitize

Resources

- ASP.NET Identity: http://www.asp.net/identity
- IdentityServer: https://github.com/IdentityServer/Thinktecture.IdentityServer3
- Pluralsight courses:
  - Web API v2 Security: http://www.pluralsight.com/courses/webapi-v2-security
  - Securing JavaScript Applications: http://www.pluralsight.com/courses/angularjs-security-fundamentals

brian.noyes@solliance.net
@briannoyes
http://briannoyes.net