Securing Web 2.0 Applications

David Brussin, Founder & CEO
Monetate, Inc.

The good old days...

- Browser
- Front-end server
- Back-end server
- Database, mainframe, etc

The good old days...

- Network security mechanisms
- Simple protocols with inspection
- Controlled scope of compromise

Security requirements

- Authentication
- Authorization
- Confidentiality
- Integrity
- Availability
- Non-repudiation
- Audit

Authentication

- Username/password auth
  - SSL for auth only?
- Persistent sessions
  - Session hijacking
- OpenID

Authorization

- Permissions, roles, capabilities
- Implicit vs. explicit authorization
  - Authorization by virtue of page layout

Confidentiality

- XSS - Cross-site Scripting
- CSRF - Cross-site Request Forgery
- Browser history and cache attacks

Integrity

- XSS - Cross-site Scripting
- CSRF - Cross-site Request Forgery
- Data pollution
- Spam accounts

Availability

- "Security is Availability in the face of Adversity"
- Traffic amplification
- 3rd party API availability and volume limits

Non-repudiation

- Typical non-repudiation today is “our system says it happened”
- Real non-repudiation requires client-side encryption
  - data signed with client certificate

Audit

- Log entries
  - Security of storage
  - Date/time synchronization
- Log injection?
- Repudiation of entries?

Secure architecture

- Trust domains
- Trust boundaries between domains
- Simple, well-defined protocols cross boundaries
- Inspection at protocol endpoints
- Resistant to individual failures

3 layers

- Architecture
- Protocols
- Applications

Ajax and user-supplied content

- Implicit vs. explicit APIs
- SQL injection & XSS
- Client-side inspection & business logic
- CSRF
  - Ajax requests as well as form posts

The mashup problem

- Dependencies on 3rd-party provided data
  - Treat 3rd-party data as user supplied
- Authentication of nodes

Multitenacy

- Multitenant hosting environments
  - Virtualization
- Multitenant applications

Scope of compromise

- What happens when things fail?
  - Browser
  - Front-end web server
  - Back-end application or database server

Frameworks making things easier?

- built in support for authentication & authorization
- clear separation of components
- plugins, middleware for
  - session anti-hijacking, anti-replay
  - CSRF

Testing

- XSS, CSRF, SQL injection, LDAP injection, HTTP header injection, XPATH injection
- wapiti.sourceforge.net
- www.watchfire.com/products/appscan

Questions?

Thank you

slides will be posted at WhatComesNext.brussin.com