Minimum Necessary Implementation: Reducing Attack Surface to Increase Security

Saturday, May 14, 2011

The Open Web Application Security Project
Long Island Chapter
Hempstead, New York

Robert Gezelter Software Consultant
35 – 20 167th Street, Suite 215
Flushing, New York 11358 – 1731
United States of America

+1 (718) 463 1079
gezelter@rlgsc.com
http://www.rlgsc.com

If it makes a sound, SET IT ON SILENT
(e.g., mobiles, pagers, PDA, smartphone)

Web Application Security

- many components
- many vulnerabilities
- What this talk does not cover:
  - Encryption
  - Communications
  - Applications Design

Today's topic:

- Attack Surface
  - Unnecessary technology vulnerabilities

A matter of trust –

- Multilateral trust is a danger
  - Scripts/applets trust browser environment
  - Client/server both trust communications
  - Servers trust clients

Poorly-placed trust is THE problem –

- Malevolent clients
- Compromised servers
- Unfaithful intermediaries

Foundation:

- "... maintain maximum security by minimizing distribution"
  – Office of Naval Operations, circa 1941
- a simple principle
  - What one does not have, cannot be leaked.
  - "Once three people know it, it is not a secret"

In Operating Systems –

- protected access modes
- “minimum necessary privilege”
- The basis of ALL multiuser environments
  - What is prohibited cannot be abused.

What does this mean for Web Applications?

- Consider different hazards/attacks
  - SQL Injection
  - Business Logic errors

A common thread:

- excessive trust/capability
  - more access/authority than needed
  - escalation

The role of technologies –

- capability vs. need

  - Different technologies
  - Different capabilities
  - Match technology, capabilities to need

Example:

- Changes in visual appearance
  - CSS: mouseover event
  - JavaScript: mouseover event
  - Java: mouseover event
  - Each has different capabilities, vulnerabilities

A Deeper Look: CSS

- Limited to stylistic issues
- Cannot be used as a springboard

A Deeper Look: JavaScript

- Full access to the DOM
- If signed: Potential full access to the machine
- A question of trust
- No Public/Private methods

A Deeper Look: Java

- Full access to the DOM
- If Signed: Potential full access to the machine
- A question of trust
- Private methods; FINAL

Different potentials

- CSS: very constrained
- Java/JavaScript: more capable; also exploitable

SQL Injection

- an example of “excessive capability”
- server excessively trusts client
  - http://xyz.com?user=fred&...
  - http://xyz.com?user=charlie&...
  - Why grant trust?

Attack surface: A matter of trust

- Minimize trust
- Up front efforts far cheaper than remediation
- VERY fat tail for compromies

Questions?

Robert Gezelter Software Consultant
35 – 20 167th Street, Suite 215
Flushing, New York 11358 – 1731
United States of America

+1 (718) 463 1079
gezelter@rlgsc.com
http://www.rlgsc.com

Session Notes & Materials:
http://www.rlgsc.com/owasp/longisland/2011/index.html