Promoting Application Security within Federal Government

Dr. Sarbari Gupta, CISSP, CISA
Founder/President
Electrosoft
sarbari@electrosoft-inc.com
703-437-9451 ext 12

AppSec DC

November 13, 2009

The OWASP Foundation

http://www.owasp.org

Agenda

- Federal IT Security Landscape
- FISMA and NIST Security Controls
- Mapping of AppSec Best Practices to Federal Specs
- Observations
- Wrap-Up

Are Federal Government Web Apps Secure?

- Well ..... *Not really!*
- Why? Weaknesses in one/more of the following:
  - Training Developers/Managers on secure development
  - Documented secure coding standards
  - Formalized SDLC processes that are actually followed
  - Effective application threat modeling
  - Security review of design/architecture
  - Security focus during code review
  - Comprehensive security testing
  - Vulnerability and Penetration Analyses
  - Active security monitoring

Information Security – Federal Compliance Landscape

- Title III of E-Government Act of 2002
  - Federal Information Security Management Act (FISMA)
- Homeland Security Presidential Directives
  - HSPD-7, HSPD-12, etc.
- OMB Memos
  - FISMA Reporting
  - Privacy
  - Data Encryption
  - FDCC, etc.)

FISMA - Security Assessment and Authorization

- NIST Standards and Guidelines
  - FIPS 199 – Security Categorization of Information Systems
  - SP 800-37 – Guidelines for Security Authorization of Information Systems
  - SP 800-53 Rev 3 – Recommended Security Controls for Federal Information Systems and Organizations

NIST Special Pub 800-53 Revision 3 (Aug '09)

- **Title**: Recommended Security Controls for Federal Information Systems and Organizations
- **Approach**: Risk Management Framework
  - Categorize Information System
  - Select Security Controls
  - Implement Security Controls
  - Assess Security Controls
  - Authorize Information System
  - Monitor Security Controls
- **18 families** of Security Controls
- Families that impact Application Security:
  - AC, AU, IA, SA, SC and SI
  - CA, PL, RA

ID	FAMILY	CLASS
AC	Access Control	Technical
AT	Awareness and Training	Operational
AU	Audit and Accountability	Technical
CA	Security Assessment and Authorization	Management
CM	Configuration Management	Operational
CP	Contingency Planning	Operational
IA	Identification and Authentication	Technical
IR	Incident Response	Operational
MA	Maintenance	Operational
MP	Media Protection	Operational
PE	Physical and Environmental Protection	Operational
PL	Planning	Management
PS	Personnel Security	Operational
RA	Risk Assessment	Management
SA	System and Services Acquisition	Management
SC	System and Communications Protection	Technical
SI	System and Information Integrity	Operational
PM	Program Management	Management

Application Security - Related Federal Efforts

- DISA
  - Application Security and Development STIG – July 2008
  - Application Security and Development Checklist Version 2 Release 1.5 - June 2009
- NIST
  - SP 800-64 Rev 2 – Security Considerations in the System Development Life Cycle – Oct 2008
  - SP 800-115 (draft) - Technical Guide to Information Security Testing – Nov 2007
  - Security Content Automation Protocol (SCAP)

OWASP Top Ten Vulnerabilities (2007)

OWASP Top Ten Vulnerabilities	NIST 800-53 Rev3 Controls
A1 - Cross Site Scripting (XSS)	SI-10: Information Input Validation
A2 - Injection Flaws	SI-10: Information Input Validation
A3 - Malicious File Execution	Not specified
A4 - Insecure Direct Object Reference	AC-3: Access Enforcement
A5 - Cross Site Request Forgery (CSRF)	Not specified
A6 - Information Leakage & Improper Error Handling	SI-11: Error Handling
A7 - Broken Authentication and Session Mgmt	SC-23: Session Authenticity
A8 - Insecure Cryptographic Storage	SC-13: Use of Cryptography
A9 - Insecure Communications	SC-9: Transmission Confidentiality
A10 - Failure to Restrict URL Access	AC-3: Access Enforcement

SANS Top 25 (1 of 3) - Insecure Interaction Between Components

Top 25 Coding Vulnerabilities	NIST 800-53 Rev3 Controls
CWE-20: Improper Input Validation	SI-10: Information Input Validation
CWE-116: Improper Encoding or Escaping of Output	Not specified
CWE-89: SQL Injection	SI-10: Information Input Validation
CWE-79: Cross-site Scripting	SI-10: Information Input Validation
CWE-78: OS Command Injection	SI-10: Information Input Validation
CWE-319: Clear-text Transmission of Sensitive Information	SC-9: Transmission Confidentiality
CWE-352: Cross-Site Request Forgery (CSRF)	Not specified
CWE-362: Race Condition	Not specified
CWE-209: Error Message Information Leak	SI-11: Error Handling

SANS Top 25 (2 of 3) – Porous Defenses

Top 25 Coding Vulnerabilities	NIST 800-53 Rev3 Controls
CWE-285: Improper Access Control (Authorization)	AC-3: Access Enforcement
CWE-327: Use of a Broken or Risky Cryptographic Algorithm	SC-13: Use of Cryptography
CWE-259: Hard-Coded Password	IA-5: Authenticator Management
CWE-732: Insecure Permission Assignment for Critical Resource	AC-3: Access Enforcement
CWE-330: Use of Insufficiently Random Values	Not specified
CWE-250: Execution with Unnecessary Privileges	AC-6: Least Privilege
CWE-602: Client-Side Enforcement of Server-Side Security	Not specified

SANS Top 25 (3 of 3) - Risky Resource Management

Top 25 Coding Vulnerabilities	NIST 800-53 Rev3 Controls
CWE-119: Memory Buffer Overrun	SA-8: Security Engineering Principles[1]
CWE-642: External Control of Critical State Data	SA-8: Security Engineering Principles[1]
CWE-73: External Control of File Name or Path	SA-8: Security Engineering Principles[1]
CWE-426: Un-trusted Search Path	SA-8: Security Engineering Principles[1]
CWE-94: Code Injection	SA-8: Security Engineering Principles[1]
CWE-494: Download of Code Without Integrity Check	SI-7: Software and Information Integrity
CWE-404: Improper Resource Shutdown or Release	SA-8: Security Engineering Principles[1]
CWE-665: Improper Initialization	SA-8: Security Engineering Principles[1]
CWE-682: Incorrect Calculation	SA-8: Security Engineering Principles[1]

1 – Weak Mapping

OWASP Application Security Verification Std 2009

ASVS Security Requirement Areas	NIST 800-53 Rev 3 Controls	Coverage
V1 - Security Architecture Documentation	RA-3	1 of 6
V2 - Authentication Verification	AC-2, AC-3, AC-5, AC-7, AC-11, AC-14, AU-2, IA-5, IA-6, IA-8, SC-24, SI-3	12 of 15
V3 - Session Management Verification	AC-11, SC-10, SC-23, SI-3	9 of 13
V4 - Access Control Verification	AC-2, AC-3, AC-6, SI-3, AU-2	10 of 15
V5 - Input Validation Verification	SA-8, SI-3, SI-10, AU-2	7 of 9
V6 - Output Encoding/Escaping Verification	SI-3, SI-10	5 of 10
V7 - Cryptography Verification	IA-5, SC-12, SC-13, SI-3, AU-2	6 of 10
V8 - Error Handling and Logging Verification	SI-3, SI-11, AU-3, AU-9	7 of 12
V9 - Data Protection Verification		0 of 6
V10 - Communication Security Verification	AC-4, AC-6, IA-3, IA-5, SC-8, SC-9, SC-24, AU-2	7 of 9
V11 - HTTP Security Verification	SC-23	1 of 7
V12 - Security Configuration Verification	CM-5, SI-6, SI-7, AU-2	3 of 4
V13 - Malicious Code Search Verification	SI-3, SI-7	2 of 2
V14 - Internal Security Verification	SC-4, SC-28	2 of 3

Fed AppSec - Weaknesses in SDLC Phases (SP 800-64)

- Phase: Initiation
  - Planning
  - System Categorization/Privacy Impact Assessment
  - Establish Secure Development Processes (Coding Stds/Config Mgmt) **WEAK!**
  - Security Training of Developer/Managers **WEAK!**
- Phase: Development
  - Risk Assessment/Threat Modeling **WEAK!**
  - Security Architecture and Documentation **WEAK!**
  - Security Testing (Design Review/Code Review/Functional & Security Testing) **WEAK!**
- Phase: Implementation/Assessment
  - Establish Secure Configuration/Environment
  - Assess System Security Posture **WEAK!**
  - Authorize System for Operation
- Phase: Operations and Maintenance
  - Perform Configuration Management of System
  - Continuous Monitoring
- Phase: Disposal
  - Planning
  - Sanitize Media/Dispose of HW & SW/Preserve Information
  - Close System

Fed AppSec– Areas to be Strengthened

- Security Training of Application Developers and Managers
- Development of Secure Coding Practices
- Threat Modeling based on Application Vulnerabilities and Threats
- Security Architecture Documentation
  - Roles/Resources/Functions/Security Controls/Components/Interactions
- Sensitive Resource Identification and Protection
  - Data, URLs, Configuration Files, etc.
- Conditioning of Output Content/Data
- Server-Side Implementation of Security Services
  - Common Implementation
  - Non-circumventable
- Secure Degradation/Failure of Functions
- Security Testing of Software Applications
  - Static and Dynamic Testing of code
  - Penetration Testing of Deployed Application

Final Thoughts

- Compliance drives federal government security
  - Current FISMA security controls and practices have greatly strengthened platform and network security
- It's time to focus on Application/Software Security
- Recommendations to promote AppSec in Federal Space:
  - Set of Security Controls for AppSec
  - Guidelines for Secure Coding Practices
  - Guidelines for Software Security Testing
  - OMB mandate to focus on AppSec