PACKETLABS

Web Application Methodology

Methodology

Our security testing methodology is derived from the OWASP Top 10:2013 and has been enhanced with current threats and our overall experience in the industry. Our methodology is comprehensive and has been broken up based on which areas can be tested with automation and those which require extensive manual testing.

Phase	Tasks Completed	Manual	Automated
Recon & Mapping	✓ Conduct search engine discovery and reconnaissance for information leakage ✓ Fingerprint web server ✓ Review web server metafiles for information leakage ✓ Enumerate applications on webserver ✓ Review webpage comments and metadata for information leakage ✓ Identify application entry points ✓ Identify technologies (e.g., web applications, frameworks or CMS platforms) used ✓ Map visible content and perform automated spidering of referenced content ✓ Test for debug parameters ✓ Discover hidden & default content	✓	✓
Discovery	**Configuration and Deploy Management Testing** ✓ Test network/infrastructure configuration ✓ Test application platform configuration ✓ Test file extensions handling for sensitive information ✓ Analyze backup and unreferenced files for sensitive information ✓ Enumerate Infrastructure and application admin interfaces ✓ Test HTTP methods ✓ Test HTTP strict transport security ✓ Test RIA cross domain policy ✓ Test for web server vulnerabilities ✓ Testing for vulnerabilities in third-party applications (e.g, Wordpress, Joomla, Drupal, Sharepoint)	✓	✓
	Identity Management Testing ✓ Test role definitions ✓ Test user registration process ✓ Test account provisioning process ✓ Testing for account enumeration and guessable user account ✓ Testing for weak or unenforced username policy ✓ Test permissions of guest/training accounts ✓ Test account suspension/resumption Process	✓	--

Phase	Tasks Completed	Manual	Automated
	Authentication Testing ✓ Testing for credentials transported over an encrypted channel ✓ Testing for default credentials ✓ Testing for weak lock out mechanism ✓ Testing for bypassing authentication schema ✓ Test remember password functionality ✓ Testing for browser cache weakness ✓ Testing for weak password policy ✓ Testing for weak security question/answer ✓ Testing for weak password change or reset functionalities ✓ Testing for weaker authentication in alternative channel		
	Authorization Testing ✓ Testing directory traversal/file include ✓ Testing for bypassing authorization schema ✓ Testing for privilege escalation ✓ Testing for insecure direct object references		--
	Session Management Testing ✓ Testing for bypassing session management schema ✓ Analyze cookies attributes (e.g., HttpOnly, Secure flags and scope) ✓ Testing for session fixation ✓ Testing for cross site request forgery ✓ Testing for logout functionality ✓ Test session timeout ✓ Testing for session puzzling ✓ Persistent cookies ✓ Test tokens for predictability ✓ Check for insecure transmission of session tokens		--
	Input Validation Testing ✓ Fuzz all input parameters ✓ Testing for reflected cross site scripting ✓ Testing for stored cross site scripting ✓ Testing for HTTP verb tampering ✓ Testing for HTTP parameter pollution ✓ Testing for HTTP splitting/smuggling ✓ Testing for SQL injection (Oracle, MySQL, MsSQL, PostgreSQL, Microsoft Access, NoSQL)		

Phase	Tasks Completed	Manual	Automated
	✓ Testing for LDAP injection ✓ Testing for ORM injection ✓ Testing for XML injection ✓ Testing for SSI injection ✓ Testing for XPath injection ✓ Testing for IMAP/SMTP injection ✓ Testing for code injection ✓ Testing for local file inclusion ✓ Testing for remote file inclusion ✓ Testing for command injection ✓ Testing for native software flaws (buffer overflow, integer bugs, format strings) ✓ Testing for incubated vulnerabilities ✓ Testing for open redirection ✓ Testing for SOAP injection		
	Error Handling ✓ Analysis of error codes ✓ Analysis of stack traces		
	Cryptography ✓ Testing for weak SSL/TLS ciphers, insufficient transport layer protection ✓ Testing for padding oracle ✓ Testing for sensitive information sent via unencrypted channels ✓ Testing for CBC bit flipping ✓ Testing for hash length extension		Partial
	Business Logic Testing ✓ Identify the logic attack surface ✓ Test business logic data validation ✓ Test ability to forge requests ✓ Test integrity checks ✓ Test for process timing (race conditions, TOCTOU) ✓ Testing for the circumvention of work flows ✓ Test defenses against application misuse ✓ Test upload of unexpected file types ✓ Test upload of malicious files ✓ Analyze SSL responses for caching of sensitive content ✓ Analyze content for sensitive data in URL parameters ✓ Testing for reliance on client-side input validation ✓ Testing of trust boundaries		--

Phase	Tasks Completed	Manual	Automated
	Client Side Testing ✓ Testing for DOM based cross site scripting ✓ Testing for JavaScript execution ✓ Testing for HTML injection ✓ Testing for client side open redirection ✓ Testing for CSS injection ✓ Testing for client side resource manipulation ✓ Test cross origin resource sharing ✓ Testing for cross site flashing ✓ Testing for clickjacking ✓ Testing WebSockets ✓ Test web messaging ✓ Test local storage ✓ Testing of thick-client components (Java, ActiveX, Flash)		Partial
Exploitation **	✓ Leverage findings from previous phases in order to to expand foothold in environment. ✓ Execute a number of exploits focusing on: ○ bypass attacks ○ injection attacks ○ session attacks ✓ Attempt to escalate privileges and/or gain unauthorized access ✓ Attempt to pivot from compromised systems to other internal systems.		--
Reporting	✓ Draft detailed report outlining findings coupled with control recommendations including executive summary outlining the overall state of the application. ✓ Document steps to reproduce findings to ensure application developers can validate remediation efforts prior to retesting. ✓ Conduct root cause analysis of findings outlining common themes observed with recommendations to improve security within the environment.		

** Packetlabs will be transparent during the execution of this phase and will review/discuss approval process in greater detail prior to the commencement of testing.