SecurityRisk ADVISORS

Application Security Testing

Our testing attempts to identify insecure web server software, application functionality and configuration settings that are susceptible to both common and custom attacks that could result in data compromise, elevated privilege or administrative application control, or reputation damage. We perform testing of the application from the perspectives of both an **unauthenticated** and **authenticated** Internet hacker. Unauthenticated testing is limited when it comes to application security testing and comprises just the first day of testing. During authenticated testing, we simulate an attacker who has registered an account or compromised an existing account within your applications.

1. Reconnaissance

We manually crawl the application as both an unauthenticated and an authenticated user in order to understand intended functionality, services, interfaces and design patterns. We conduct advanced searching and mining techniques to attempt to enumerate application and user information that may contribute to potential application weaknesses.

We perform application discovery to analyze hosts, URLs and services. During this step we use port scanning and other reconnaissance techniques to profile the application and its components, enumerate version information, and compile a footprint of resources to conduct further analysis of potential vulnerabilities impacting the application and supporting infrastructure.

2. Research, Verify & Assess

We use a combination of manual and automated testing techniques to first identify common web application security weaknesses, such as those described in the Open Web Application Security Project (OWASP) Top Ten. Our manual web application testing also focuses on identifying unique application vulnerabilities and business logic flaws that could allow unauthorized access to sensitive information or administrative functionality. Automated tools assist in detecting pervasive common security vulnerabilities at a high speed, but with a trade-off in accuracy. We validate and eliminate false positives for vulnerabilities identified through automated testing.

3. Evidence & Recommend

We sort and validate potential false positives, and we present appropriate evidence of our access though screen captures and/or sample data to illustrate and explain the impact to the business for each of our observations. We categorize our observations by risk in a detailed observations matrix, and provide recommendations to address any vulnerabilities that we might find. In addition to short term fixes, we describe the process-oriented weaknesses and recommend changes designed to prevent the issues from reappearing.

Contact Us

Phone: 215.867.9051
Email: info@securityriskadvisors.com
Website: www.securityriskadvisors.com

Common Vulnerabilities

Input Validation Flaws

Cross-Site Scripting (XSS), SQL injection, XML injection and other injection flaws that can allow data access or victim takeover.

Cross-Site Request Forgery

Provide an attacker the ability to craft forged HTTP/S requests and trick an authenticated user into carrying out the transaction.

Broken Access Control

Ability to access another user's information or administrative functionality without proper authorization, unintended access to application URLs and service endpoints.

Security Misconfigurations

Broad category that can include default vendor configurations, development/test/debug functionality, configurations in third party code, default account passwords, and unpatched software.

Insecure Authentication Mechanisms

Weaker authentication schemes, ability to bypass primary or secondary forms of authentication (e.g. password self-service), or weaknesses in the implementation of resource or service-level authentication.

Improper Error Handling

Verbose information returned to an untrusted client. This can take the form of unhandled server errors, application or service-level exceptions encountered, client side logging, and other information leakage scenarios.

Unrestricted URL Redirection

Lack of proper URL validation that can allow an attacker to redirect or forward a user to a malicious site, most often used in a phishing attack.

Insufficient Transport Layer Security

Insecure requests between client and server, such as the lack of HTTPS to transmit user credentials and session identifiers, and presence of insecure cipher suites and TLS vulnerabilities.

Session Management Weaknesses

Session management implementation susceptible to hijacking attacks, lack of proper expiration and timeout, or scope of access too broadly-defined.

Insufficient cryptographic protection

Sensitive data cached in clear-text on the client, insecure storage and handling of account credentials and other sensitive information within the application and backend components.

Business logic flaws

Design flaws and coding errors that lead to unintended application behavior or transactions that can be manipulated for profit or data loss.