secarma
CYBERSECURITY EXPERTS

A Advise C Certify T Test

Web Application Penetration Testing

As a direct interface with clients, applications are usually designed with functionality and aesthetics in mind with security considerations coming in second place.

However, web application security risks can be significant, with a range of issues including confidential data exposure or brand damage through public attacks.

Who Is It For?

Almost all modern organisations will have some type of bespoke web application, from simple brochureware sites advertising their services, to bespoke applications that integrate business-specific logic.

We offer security testing appropriate for all levels of complexity, from simple security reviews of Content Management Systems, to deep-dive assessments of bespoke applications.

We often engage directly with development teams who are conscious of building security into the fundamental design of their application, but also with end users who are looking for assurances about the software they are using.

How Can We Help?

From remote application tests to on-site, detailed investigations, our application security assessment services are individually tailored to your needs, delivered by penetration testers who specialise in security at the application layer.

Application testing can include an approach which aims to replicate the approach an external attacker would take, or testing can be fully informed, such as including documentation or code-assisted techniques to ensure a more efficient approach.

What We Test

From common vulnerabilities to complex application logic, our methodology includes but is not limited to, the OWASP Top 10. For example, testing for application issues such as:

- **Identity and Access Management**
  Ensuring accounts utilise multifactor authentication and adhere to the principle of least privilege.
- **Application logic**
  Abuse of functionality and logical flaws within applications.
- **Authentication attacks**
  Username enumeration, brute force attacks, and credential stuffing.
- **Authorisation**
  Insufficient credential and session management.
- **Client-side Attacks**
  Cross-site scripting and response splitting.
- **Command Execution**
  Injection attacks, deserialization and buffer overflow flaws.
- **Insecure File Upload**
  Insecure handling of uploaded files allowing code execution, cross-site scripting, or sensitive data exposure.

3 Archway, Birley Fields, Manchester M15 5QJ
0161 513 0960
secarma.com
actnow@secarma.com

Our web application testing methodology is aligned with the OWASP Top 10, but not limited to this list alone. A list of the major testing categories we will perform is outlined below:

- **Information Gathering**

  Before the engagement begins, we will map the attack surface to discover alive hosts, services, and versions. This can include:

  Open Source Intelligence - Online sources of information will be used to gather intelligence about the target organization, such as social media sites.

  Application Mapping - Spidering and forced browsing techniques are used to discover application functionality, including both linked and unlinked content.

  Directory Indexing - Where directory indexing is enabled additional information about the target system and application will be exposed.

  Information Leakage and Verbose Errors - Applications that disclose information unintentionally such as through verbose error messages, will be leveraged to gain more insight into technologies and configuration options in use.

- **Proof Of Concept And Confirmation**

  Where vulnerabilities are discovered a proof of concept exploit will be created to demonstrate the potential business risk. This ensures that false positives are removed by manually confirming and demonstrating all discovered vulnerabilities.

- **Exploitation**

  Exploitation involves discovering weaknesses within exposed applications and leveraging those weaknesses.

  Business Logic Flaws - We attempt to bypass the expected logic flow of the application to demonstrate risk.

Injection - We test for issues such as SQL, Command, and LDAP injection, to bypass authentication or extract data.

Authentication - Broken Authentication includes issues such as weak session management flaws, lack of bruteforce protection, and lack of credential stuffing protection.

Sensitive Data Exposure - Unintentionally exposing confidential information, lack of transport security, or insecure data storage.

XML External Entity Injection - Entity injection can allow access to sensitive files, internal file shares, or command execution.

Broken Access Control - Issues that allow for access control bypasses or issues such as missing functional level access control.

Security Misconfiguration - Common and default misconfigurations allow for information exposure or include lack of application hardening through security headers.

Cross-site Scripting (XSS) - XSS attacks allow for a range of attacks such as defacements, confidential data theft, or distribution of malicious software.

Insecure Deserialization - Deserialization can often allow for code execution and privilege escalation attacks.

Known Vulnerabilities - Vulnerabilities in software dependencies, libraries and modules.

Request Forgery - Issues such as cross-site request forgery and server-side request forgery are assessed.

Insecure Redirects - Unvalidated redirects can allow for more advanced social engineering attacks such as phishing.

The assessment can include both unauthenticated and authenticated assessments, to demonstrate the risks of an opportunistic attacker without any access, as well as a rogue user.