**Information Gathering/Recon**

* Retrieve and analyze the robot.txt files by using a tool called GNU Wget.
* Examine the version of the software/database details, the error technical component, and bugs by the error codes and by requesting invalid pages
* Discover hidden and default content
* Perform directory style searching and vulnerability scanning and probe for URLs, using tools such as NMAP and Nessus.
* By using a traditional Fingerprint Tool, such as Nmap, Amap, perform TCP/ICMP, and service Fingerprinting.
* Identify the entry point of the application using OWSAP ZAP, Burb Proxy, TemperIE, and WebscarabTemper Data.
* Identify the technologies used
* Map the attack surface

**Test Handling of Access**

* Authentication
* Test password quality rules
* Test for username enumeration
* Test resilience to password guessing
* Test any account recovery function
* Test any “remember me” function
* Test username uniqueness
* Check for unsafe distribution of credentials
* Session handling
* Test tokens for meaning
* Test tokens for predictability
* Check for insecure transmission of tokens
* Check for disclosure of tokens in logs
* Check the map of tokens to sessions
* Check session termination
* Check for session fixation
* Check for cross-site request forgery
* Check cookie scope
* Access controls
* Understand the access control requirements
* Test effectiveness of controls, using multiple accounts if possible
* Test for insecure access control methods (request parameters, Referer header, etc.)

**Test Handling of Input**

* Fuzz all request parameters
* Test for SQL injection
* Identify all reflected data
* Test for reflected XSS
* Test for HTTP header injection
* Test for arbitrary redirection
* Test for stored attacks
* Test for OS command injections
* Test for path traversal
* Test for script injection
* Test for file inclusion
* Test for SMTP injection
* Test for native software flaws (buffer overflow, integer bugs, format strings)
* Test for SOAP injection
* Test for LDAP injection
* Test for XPath injection

**Test Application Logic**

* Identify the logic attack surface
* Test transmission of data via the client
* Test for reliance on client-side input validation
* Test any thick-client components (Java, ActiveX, Flash)
* Test multi-stage processes for logic flaws
* Test handling of incomplete input
* Test trust boundaries
* Test transaction logic

**Assess Application Hosting**

* Test segregation in shared infrastructures
* Test segregation between ASP-hosted applications
* Test for web server vulnerabilities
* Default credentials
* Default content
* Dangerous HTTP methods
* Proxy functionality
* Virtual hosting misconfiguration
* Bugs in web server software

**Miscellaneous Tests**

* Check for DOM-based attacks
* Check for frame injection
* Check for local privacy vulnerabilities
* Persistent cookies
* Caching
* Sensitive data in URL parameters
* Forms with autocomplete enabled
* Follow up any information leakage
* Check for weak SSL ciphers