Web Application Pentesting

cuu duong than cong . com

Objectives

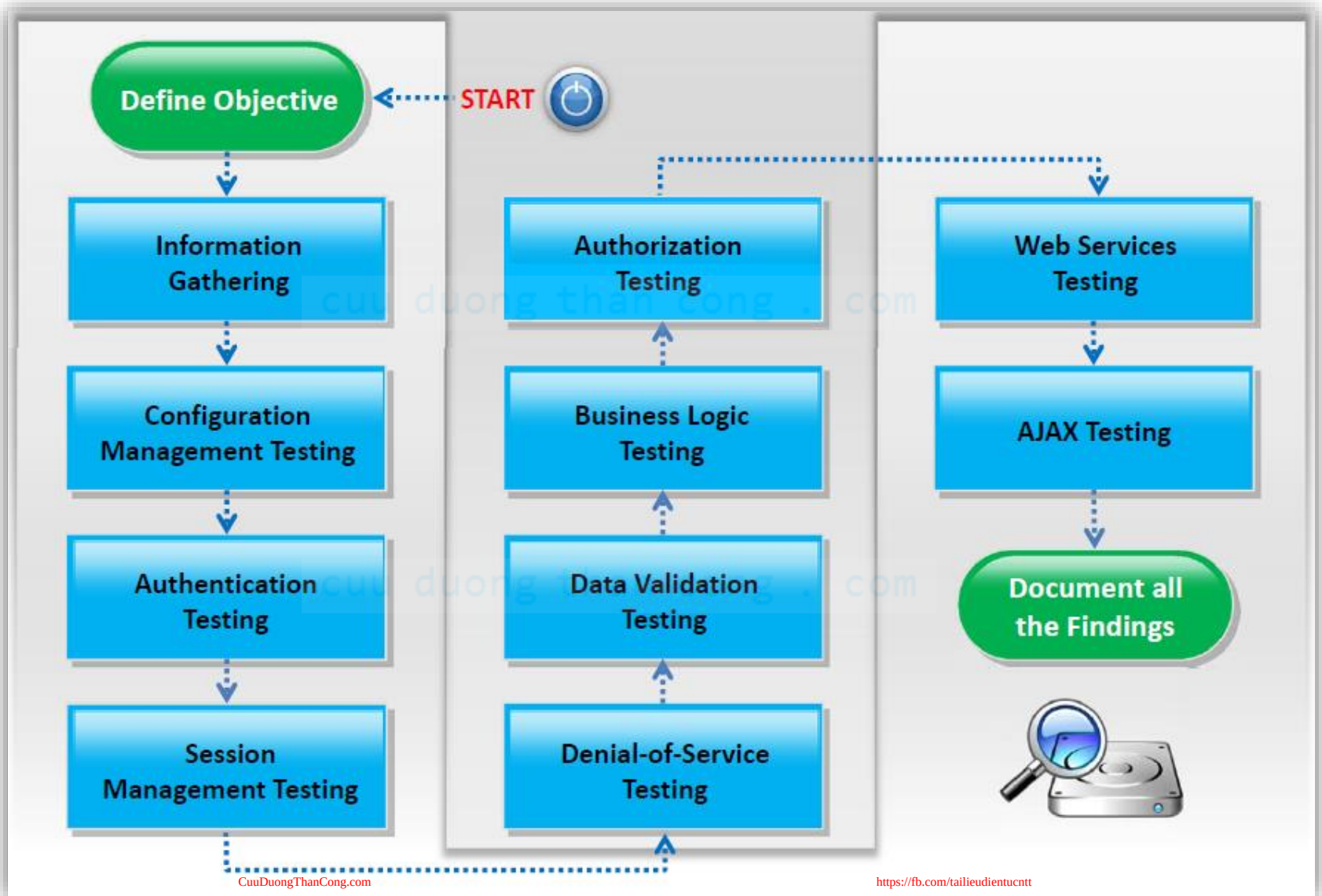
- Understand basic phases of Web application pentesting

Web application Pentesting framework

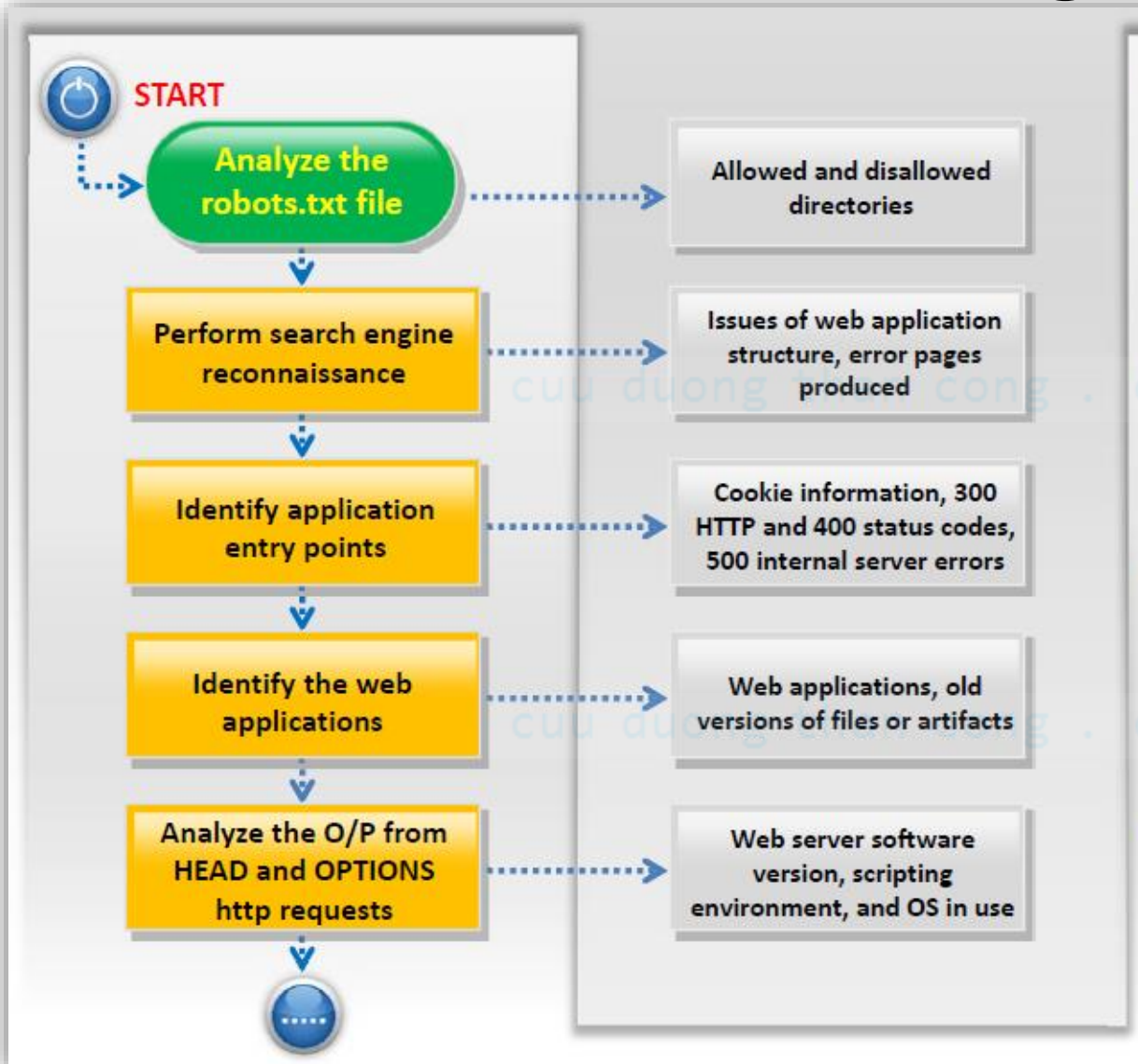
OWASP

https://www.owasp.org/index.php/Web_Application_Penetration_Testing

Web Application Pentesting

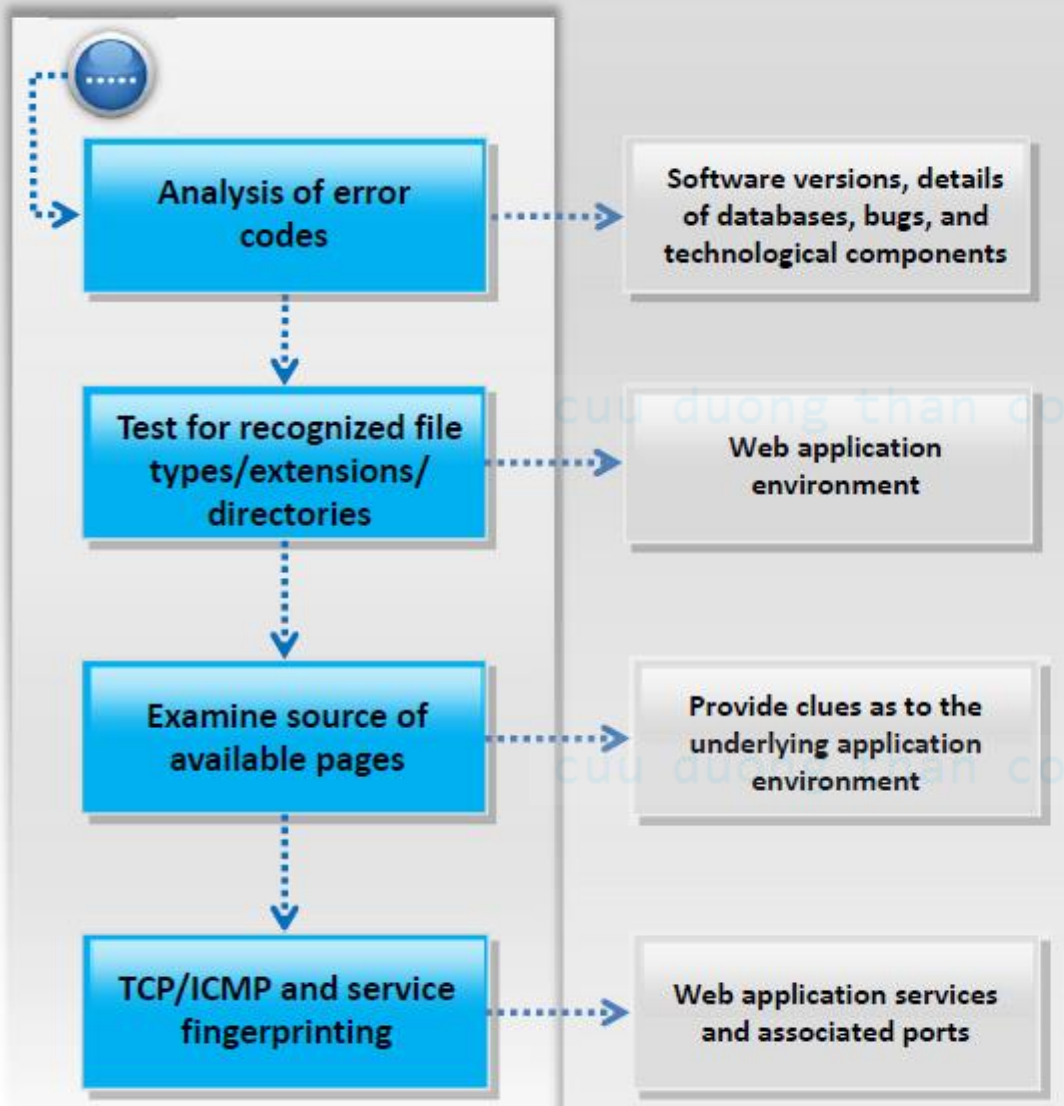


Information Gathering



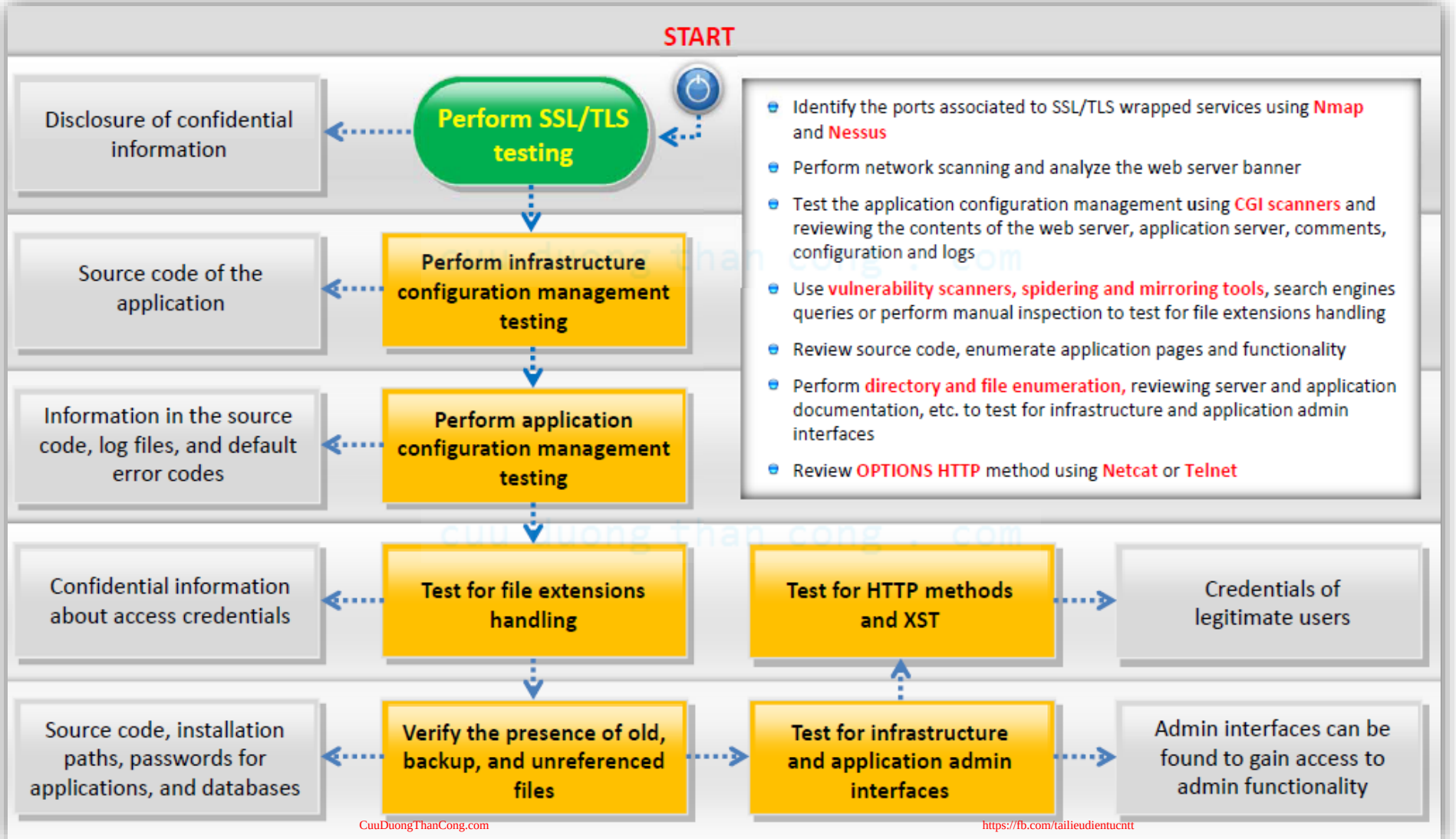
- Retrieve and analyze robots.txt file using tools such as **GNU Wget**
- Use the advanced **"site:" search operator** and then click **"Cached"** to perform search engine reconnaissance
- Identify application entry points using tools such as **Webscarab, Burp proxy, OWASP ZAP, TamperIE** (for Internet Explorer), or **Tamper Data** (for Firefox)
- To identify web applications: probe for URLs, do dictionary-style searching (intelligent guessing) and perform vulnerability scanning using tools such as **Nmap** (Port Scanner) and **Nessus**
- Implement techniques such as DNS zone transfers, DNS inverse queries, web-based DNS searches, querying search engines (googling)

Information Gathering

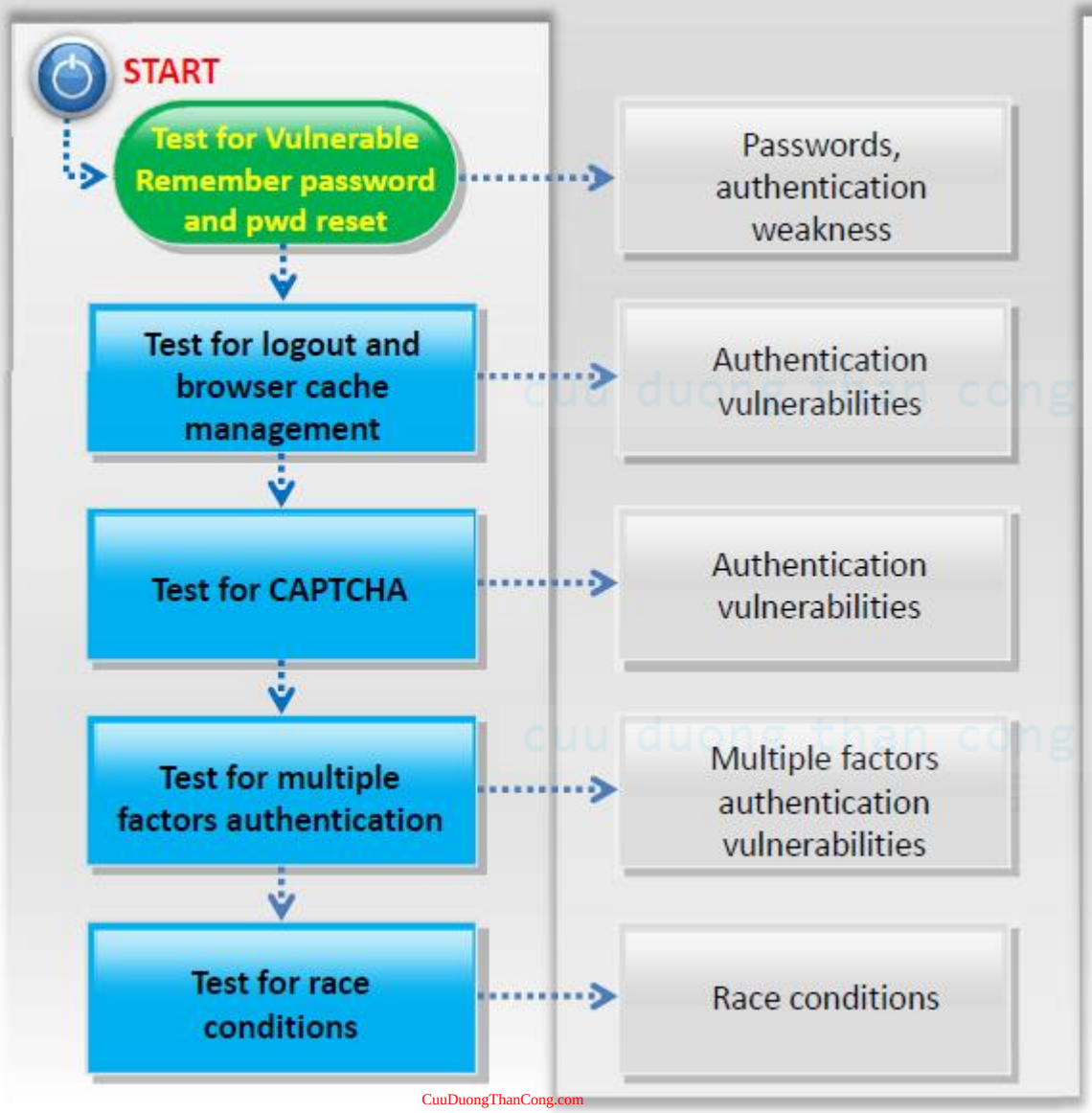


- Analyze error codes **by requesting invalid pages** and **utilize alternate request methods** (POST/PUT/Other) in order to collect confidential information from the server
- Examine the source code from the accessible pages **of the application front-end**
- Test for recognized file types/extensions/directories by requesting common file extensions such as .ASP, .HTM, .PHP, .EXE, and **watch for any unusual output or error codes**
- Perform TCP/ICMP and service fingerprinting using traditional fingerprinting tools such as **Nmap** and **Queso**, or the more recent application fingerprinting tool **Amap**

Configuration management Testing

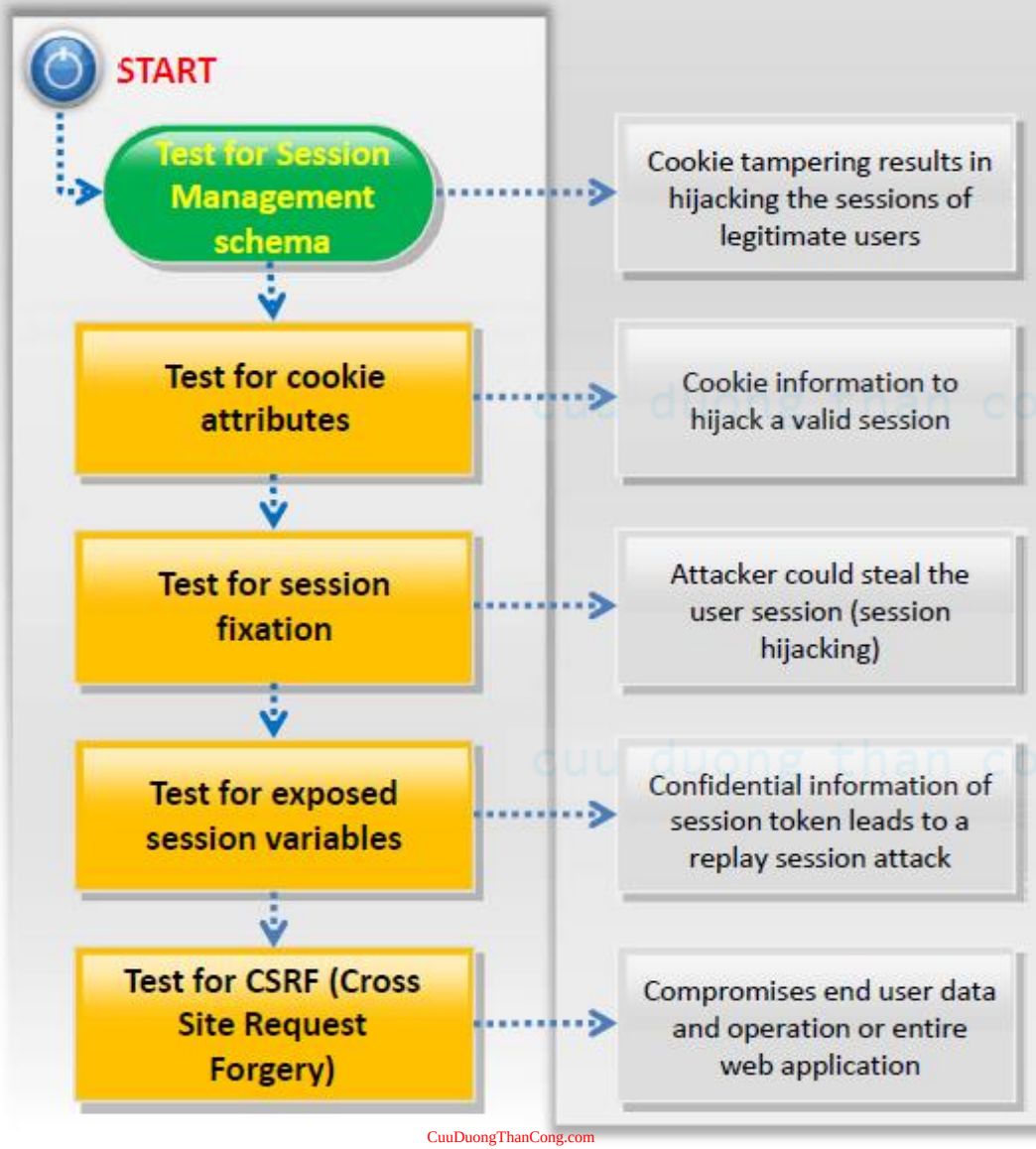


Authentication Testing



- Try to **reset passwords** by guessing, social engineering, or cracking secret questions, if used. Check if **"remember my password" mechanism** is implemented by checking the HTML code of the login page.
- Check if it is possible to **"reuse" a session after logout**. Also check if the **application automatically logs out a user** when that user has been idle for a certain amount of time, and that no sensitive data remains stored in the browser cache.
- Identify all parameters that are sent in addition to the **decoded CAPTCHA** value from the client to the server and try to send an **old decoded CAPTCHA value with an old CAPTCHA ID of an old session ID**
- Check if users hold a hardware device of some kind in addition to the password. Check if **hardware device communicates directly and independently** with the authentication infrastructure using an additional communication channel.
- **Attempt to force a race condition**, make multiple simultaneous requests while observing the outcome for unexpected behavior. Perform code review.

Session Management Testing



- Collect sufficient number of cookie samples, analyze the cookie generation algorithm and **forge a valid cookie** in order to perform the attack
- Test for cookie attributes using intercepting proxies such as **Webscarab**, **Burp proxy**, **OWASP ZAP**, or traffic intercepting browser plug-in's such as **"TamperIE"**(for IE) and **"Tamper Data"**(for Firefox)
- To test for session fixation, **make a request to the site** to be tested and analyze vulnerabilities using the **WebScarab** tool
- Test for exposed session variables by inspecting **encryption & reuse of session token,** proxies & caching , GET & POST, and transport vulnerabilities
- Examine the **URLs in the restricted area** to test for CSRF

Sample reports

PENETRATION TEST REPORT – MEGACORP ONE

activity

http://www.pentest-standard.org/index.php/Reporting

Demo

cuu duong than cong . com