# Security Assessment Checklist

* Input Validation – SQLi, XSS
* Session Manipulation
* Checking for IDOR vulnerabilities
* Path traversal/ File Inclusion
* Checking for OS command injection (Commands and SQL)
* Weak encryption algorithms
* Sensitive information and directories (Information leakage)
* Looking for sensitive files such as robots.txt
* Performing DNS queries, lookups etc.
* Weak Passwords
* Remember me
* Invalid captcha
* Misconfigurations
* Authentication Testing – testing for logging, remember my password, weak security questions and answers, authentication bypass
* Authorization Testing – Path traversal, cookie tampering, HTTP request tampering
* Identifying versions and channels
* Error handling and information disclosure
* Business Logic testing

# Tools Deployment

* Burpsuite pro
* Gobuster
* Ffuf
* Dirbuster
* Nmap
* Dnsrecon
* Sublist3r
* Domain.sh
* Dig
* Host
* Dnsenum
* Fierce
* Dnsmap

# Step-by-step procedure for testing

1. SQL Injection
   1. Create a list of target URLs or domains of focus
   2. Write a custom script to automate SQLi checks
      1. Read target URLs from the list
      2. For each URL, utilize tools to check for SQLi vulnerabilities (tools such as SQLmap can do this)
      3. Confirm the vulnerability by using different SQLi payloads and checking the response for specific errors or behavior.
   3. Execute the script and analyze the behaviour
2. Cross-Site Scripting (XSS)
   1. Create a list
   2. Write a custom script to automate checks
      1. Read target URLs from the list
      2. For each URL, run tools to automate checks for this vulnerability
      3. Confirm the vulnerability by using different payloads and analyze the result
   3. Execution
3. Insecure Direct Object Reference (IDOR)
   1. Using the same list from before or creating a new list; write a custom script to automate checks:
      1. Read targets from the list
      2. For each URL, start by changing the parameter and checking if this exposes data that should be restricted
      3. Confirm by verifying unauthorized access to these data
   2. Execution
4. Command Injection
   1. Using the same list from before or creating a new list; write a custom script to automate checks:
      1. Read targets from the list
      2. For each URL, inject OS command payloads to check for command execution.
      3. Confirm the vulnerability by checking the response for command execution indicators or changes in response time
   2. Execution