Web Application Penetration Testing

This will not include passive information gathering

Nikto | Web Server Vulnerability Scanning	
Command	**Usage**
nikto -h *TARGET*	Overall web vuln scan
nikto -h *TARGET* -o nikto.html -Format html	Export output as HTML

GoBuster | File & Directory Enumeration	
Command	**Usage**
gobuster dir -u *TARGET* -w *wordlist*	File & Directory Enumeration
gobuster dir -u *TARGET* -w *wordlist* -b *403,404*	File & Directory enumeration filtered by HTTP result code
gobuster dir -u *TARGET* -w *wordlist* -x *.php,.txt*	File & Directory enumeration filtered by file extensions
gobuster dir -u *TARGET* -w *wordlist* -r	File & Directory Enumeration + Follow the HTTP redirect if there is one

OWASP Amass | Automated Recon | GitHub	
Command	**Usage**
amass enum -d *TARGET*	Subdomain Enum
amass enum -passive -d *TARGET*	Passive Subdomain Enum
amass enum -passive -d *TARGET* -src -dir *PathToOutputFolder*	Provides from which src it found the data and saves output to specified dir

WPScan	
Command	**Usage**
wpscan --url *TARGET*	Basic enumeration
wpscan --url *TARGET* --enumerate p --plugins-detection aggressive	Plugins enum
wpscan --url *TARGET* --enumerate p --plugins-detection aggressive --api-token *APIKEY*	Plugins enum + vuln scanner

OWASP ZAP | SQLi	
Command	**Usage**
• Open OWASP ZAP and click on a browser icon (Top Mid) • Launch the target website in the browser instance • In “🌍 Sites”, Right click on the API request with the parameter you want to try a SQLi on, then click ‘Attack’ & ‘Fuzz’. • Edit → Remove the Parameter value → Save • Select the empty space where the parameter value belongs and click on ‘Add’ • Choose type: File Fuzzers → jbrofuzz → SQL Injection → Add → OK → Start Fuzzer • Check for ‘Reflected’ in the state column of the Fuzzer tab.	Automated SQL injection using a predefined list of payloads.

SQLMap	
Command	**Usage**
sqlmap -u *TARGETURL* --data “*PARAMETERNAME*” -p *PARAMETERNAME* --method *POST/GET*	Trying Injection
sqlmap -r *REQUESTFILE* -p *PARAMETERNAME* --technique=*E*	• Do an SQLMap scan using a request file. A request file is basically the API request that you caught using burpsuite. Save it in a file. • The technique means: Error based (E), Boolean based, time based, union etc…
sqlmap -r *REQUESTFILE* -p *PARAMETERNAME* --technique=*E* --current-db	Lists the current DB type, name and version
sqlmap -r *REQUESTFILE* -p *PARAMETERNAME* --technique=*E* -D *DBNAME* --tables	Lists the tables inside a DB
sqlmap -r *REQUESTFILE* -p *PARAMETERNAME* --technique=*E* -D *DBNAME* -T *TABLENAME* --dump	Dumps table data
sqlmap -r *REQUESTFILE* -p *PARAMETERNAME* --technique=*E* --current-user	Identifies through which user we have access