

@SoniBhaskar














PENTRATION TESTING

A person is seen from behind, sitting at a desk in a dimly lit room. They are looking at a large computer monitor that displays a vibrant, pixelated cityscape at night with many lights. The person has dark hair and is wearing a dark t-shirt. On the desk, there are two keyboards and a mouse. To the right of the person is a tall, dark server rack. The overall atmosphere is mysterious and tech-oriented.

BEGINNERS
TO
EXPERT!

We Are Indian We Are Great

Table of Content

-  Phase 1: History
-  Phase 2: Web and Server Technology
-  Phase 3: Setting Up the Lab with BurpSuite and bWAPP
-  Phase 4: Mapping the Application and Attack Surface
-  Phase 5: Understanding and Exploiting OWASP Top 10 Vulnerabilities
-  Phase 6: Session Management Testing
-  Phase 7: Bypassing Client-Side Controls
-  Phase 8: Attacking Authentication/Login
-  Phase 9: Attacking Access Controls
-  Phase 10: Attacking Input Validations
-  Phase 11: Generating and Testing Error Codes
-  Phase 12: Weak Cryptography Testing
-  Phase 13: Business Logic Vulnerability

Note: Some links may lead to 404 errors in the future. Please report any broken links in the issue section so we can provide replacements.

Phase 1 – History

YouTube Videos:

History of the Internet: [\[Link\]](#)

Blogs:

A Brief History of the Internet: [\[Link\]](#)

History of the Internet: [\[Link\]](#)

Books:

- Where Wizards Stay Up Late: The Origins of the Internet
- How the Internet Happened: From Netscape to the iPhone

Documentaries:

- "Explained: The Internet" (Netflix)



Bhaskar Soni

Phase 2 – Web and Server Technology

YouTube Videos:

Basic concepts of web applications, how they work and the HTTP protocol: [\[Link\]](#)

HTML basics part 1: [\[Link\]](#)

HTML basics part 2: [\[Link\]](#)

Difference between static and dynamic website: [\[Link\]](#)

HTTP protocol Understanding: [\[Link\]](#)

Parts of HTTP Request: [\[Link\]](#)

Parts of HTTP Response: [\[Link\]](#)

Various HTTP Methods: [\[Link\]](#)

Understanding URLs: [\[Link\]](#)

Intro to REST: [\[Link\]](#)

HTTP Request & Response Headers: [\[Link\]](#)

What is a cookie: [\[Link\]](#)



Bhaskar Soni

Phase 2 – Web and Server Technology [Cont.]

HTTP Status codes: [\[Link\]](#)

HTTP Proxy: [\[Link\]](#)

Authentication with HTTP: [\[Link\]](#)

HTTP basic and digest authentication: [\[Link\]](#)

What is “Server-Side”: [\[Link\]](#)

Server and client side with example: [\[Link\]](#)

What is a session: [\[Link\]](#)

Introduction to UTF-8 and Unicode: [\[Link\]](#)

URL encoding: [\[Link\]](#)

HTML encoding: [\[Link\]](#)

Base64 encoding: [\[Link\]](#)

Hex encoding & ASCII: [\[Link\]](#)



Bhaskar Soni

Phase 2 – Web and Server Technology [Cont.]

Books:

- HTTP: The Definitive Guide
- Web Application Security: Exploitation and Countermeasures for Modern Web Applications

Documentaries:

- "The Great Hack" (Netflix)
- "Code: Debugging the Gender Gap" (Netflix)



Phase 3 – Setting up the lab with BurpSuite and bWAPP

Setup lab with bWAPP: [\[Link\]](#)

Set up Burp Suite : [\[Link\]](#)

Configure Firefox and add certificate: [\[Link\]](#)

Mapping and scoping website: [\[Link\]](#)

Spidering: [\[Link\]](#)

Active and passive scanning: [\[Link\]](#)

Scanner options and demo: [\[Link\]](#)

Introduction to password security: [\[Link\]](#)

Intruder: [\[Link\]](#)

Intruder attack types: [\[Link\]](#)

Payload settings: [\[Link\]](#)

Intruder settings: [\[Link\]](#)

Other Labs:

No.1 Penetration testing tool: [\[Link\]](#)



Bhaskar Soni

Phase 3 – Setting up the lab with BurpSuite and bWAPP [Cont.]

Environment Setup: [\[Link\]](#)

General concept: [\[Link\]](#)

Proxy module: [\[Link\]](#)

Repeater module: [\[Link\]](#)

Target and spider module: [\[Link\]](#)

Sequencer and scanner module: [\[Link\]](#)

Books:

- Burp Suite Essentials
- The Web Application Hacker's Handbook



Bhaskar Soni

Phase 4 – Mapping the application and attack surface

Spidering: [\[Link\]](#)

Mapping application using robots.txt: [\[Link\]](#)

Discover hidden contents using dirbuster: [\[Link\]](#)

Dirbuster in detail: [\[Link\]](#)

Discover hidden directories and files with intruder: [\[Link\]](#)

Directory bruteforcing 1: [\[Link\]](#)

Directory bruteforcing 2: [\[Link\]](#)

Identify application entry points: [\[Link\]](#)

Identify application entry points: [\[Link\]](#)

Identify client and server technology: [\[Link\]](#)

Identify server technology using banner grabbing (telnet): [\[Link\]](#)

Identify server technology using httprecon: [\[Link\]](#)

Pentesting with Google dorks Introduction: [\[Link\]](#)



Bhaskar Soni

Phase 4 – Mapping the application and attack surface [Cont.]

Fingerprinting web server: [\[Link\]](#)

Use Nmap for fingerprinting web server: [\[Link\]](#)

Review webs servers metafiles for information leakage: [\[Link\]](#)

Enumerate applications on web server: [\[Link\]](#)

Identify application entry points: [\[Link\]](#)

Map execution path through application: [\[Link\]](#)

Fingerprint web application frameworks: [\[Link\]](#)



Phase 5 – Understanding and exploiting OWASP top 10 vulnerabilities

A closer look at all owasp top 10 vulnerabilities:

[\[Link\]](#)

Injection: [\[Link\]](#)

Broken authentication and session management:

[\[Link\]](#)

Cross-site scripting: [\[Link\]](#)

Insecure direct object reference: [\[Link\]](#)

Security misconfiguration: [\[Link\]](#)

Sensitive data exposure: [\[Link\]](#)

Missing functional level access controls: [\[Link\]](#)

Cross-site request forgery: [\[Link\]](#)

Using components with known vulnerabilities:

[\[Link\]](#)

Unvalidated redirects and forwards: [\[Link\]](#)



Bhaskar Soni

Phase 6 – Session management testing

Bypass authentication using cookie manipulation:

[\[Link\]](#)

Cookie Security Via httponly and secure Flag - OWASP: [\[Link\]](#)

Penetration testing Cookies basic: [\[Link\]](#)

Session fixation 1: [\[Link\]](#)

Session fixation 2: [\[Link\]](#)

Session fixation 3: [\[Link\]](#)

Session fixation 4: [\[Link\]](#)

CSRF - Cross site request forgery 1: [\[Link\]](#)

CSRF - Cross site request forgery 2: [\[Link\]](#)

CSRF - Cross site request forgery 3: [\[Link\]](#)

CSRF - Cross site request forgery 4: [\[Link\]](#)

CSRF - Cross site request forgery 5: [\[Link\]](#)

Session puzzling 1: [\[Link\]](#)

Admin bypass using session hijacking: [\[Link\]](#)



Bhaskar Soni

Phase 7 – Bypassing client-side controls

What is hidden forms in HTML: [\[Link\]](#)

Bypassing hidden form fields using tamper data: [\[Link\]](#)

Bypassing hidden form fields using Burp Suite: [\[Link\]](#)

Changing price on eCommerce website using parameter tampering: [\[Link\]](#)

Understanding cookie in detail: [\[Link\]](#)

Cookie tampering with tamper data: [\[Link\]](#)

Cookie tamper part 2: [\[Link\]](#)

Understanding referer header in depth using Cisco product: [\[Link\]](#)

Introduction to ASP.NET viewstate: [\[Link\]](#)

ASP.NET viewstate in depth: [\[Link\]](#)

Analyse sensitive data in ASP.NET viewstate: [\[Link\]](#)



Bhaskar Soni

Phase 7 – Bypassing client-side controls [Cont.]

Cross-origin-resource-sharing explanation with example: [\[Link\]](#)

CORS demo 1: [\[Link\]](#)

CORS demo 2: [\[Link\]](#)

Security headers: [\[Link\]](#)

Security headers 2: [\[Link\]](#)



Phase 8 – Attacking authentication /login

Brute-force login panel: [\[Link\]](#)

Username enumeration: [\[Link\]](#)

Username enumeration with bruteforce password attack: [\[Link\]](#)

Authentication over insecure HTTP protocol: [\[Link\]](#)

Authentication over insecure HTTP protocol: [\[Link\]](#)

Forgot password vulnerability - case 1: [\[Link\]](#)

Forgot password vulnerability - case 2: [\[Link\]](#)

Login page autocomplete feature enabled: [\[Link\]](#)

Testing for weak password policy: [\[Link\]](#)

Test for credentials transportation using SSL/TLS certificate: [\[Link\]](#)

Basics of MySQL: [\[Link\]](#)

Testing browser cache: [\[Link\]](#)



Bhaskar Soni

Phase 8 – Attacking authentication /login [Cont.]

Bypassing login panel -case 1: [\[Link\]](#)

Bypass login panel - case 2: [\[Link\]](#)



Bhaskar Soni

Phase 9: Attacking Access Controls

Finding admin panel: [\[Link\]](#)

Finding admin panel and hidden files and directories: [\[Link\]](#)

Finding hidden webpages with dirbusater: [\[Link\]](#)

IDOR case 1: [\[Link\]](#)

IDOR case 2: [\[Link\]](#)

IDOR case 3: [\[Link\]](#)

What is privilege escalation: [\[Link\]](#)

Privilege escalation - Hackme bank - case 1: [\[Link\]](#)

Privilege escalation - case 2: [\[Link\]](#)



Phase 10: Attacking Input Validations

HTTP verb tampering: [\[Link\]](#)

HTTP verb tampering demo: [\[Link\]](#)

HTTP parameter pollution: [\[Link\]](#)

HTTP parameter pollution demo 1: [\[Link\]](#)

HTTP parameter pollution demo 2: [\[Link\]](#)

HTTP parameter pollution demo 3: [\[Link\]](#)

XSS - Cross site scripting:

Introduction to XSS: [\[Link\]](#)

What is XSS: [\[Link\]](#)

Reflected XSS demo: [\[Link\]](#)

XSS attack method using burpsuite: [\[Link\]](#)

XSS filter bypass with Xenotix: [\[Link\]](#)

Reflected XSS filter bypass 1: [\[Link\]](#)

Reflected XSS filter bypass 2: [\[Link\]](#)



Bhaskar Soni

Phase 10: Attacking Input Validations [Cont.]

Reflected XSS filter bypass 3: [\[Link\]](#)

Reflected XSS filter bypass 4: [\[Link\]](#)

Reflected XSS filter bypass 5: [\[Link\]](#)

Reflected XSS filter bypass 6: [\[Link\]](#)

Reflected XSS filter bypass 7: [\[Link\]](#)

Reflected XSS filter bypass 8: [\[Link\]](#)

Reflected XSS filter bypass 9: [\[Link\]](#)

Introduction to Stored XSS: [\[Link\]](#)

Stored XSS 1: [\[Link\]](#)

Stored XSS 2: [\[Link\]](#)

Stored XSS 3: [\[Link\]](#)

Stored XSS 4: [\[Link\]](#)

Stored XSS 5: [\[Link\]](#)

SQL injection:

Part 1 - Install SQLi lab: [\[Link\]](#)



Bhaskar Soni

Phase 10: Attacking Input Validations [Cont.]

Part 2 - SQL lab series: [\[Link\]](#)

Part 3 - SQL lab series: [\[Link\]](#)

Part 4 - SQL lab series: [\[Link\]](#)

Part 5 - SQL lab series: [\[Link\]](#)

Part 6 - Double query injection: [\[Link\]](#)

Part 7 - Double query injection cont: [\[Link\]](#)

Part 8 - Blind injection boolean based: [\[Link\]](#)

Part 9 - Blind injection time based: [\[Link\]](#)

Part 10 - Dumping DB using outfile: [\[Link\]](#)

Part 11 - Post parameter injection error based: [\[Link\]](#)

Part 12 - POST parameter injection double query based: [\[Link\]](#)

Part 13 - POST parameter injection blind boolean and time based: [\[Link\]](#)



Bhaskar Soni

Phase 10: Attacking Input Validations [Cont.]

Part 14 - Post parameter injection in UPDATE query: [\[Link\]](#)

Part 15 - Injection in insert query: [\[Link\]](#)

Part 16 - Cookie based injection: [\[Link\]](#)

Part 17 - Second order injection: [\[Link\]](#)

Part 18 - Bypassing blacklist filters - 1: [\[Link\]](#)

Part 19 - Bypassing blacklist filters - 2: [\[Link\]](#)

Part 20 - Bypassing blacklist filters - 3: [\[Link\]](#)

Part 21 - Bypassing WAF: [\[Link\]](#)

Part 22 - Bypassing WAF - Impedance mismatch: [\[Link\]](#)

Part 23 - Bypassing addslashes - charset mismatch: [\[Link\]](#)

NoSQL injection:

Introduction to NoSQL injection: [\[Link\]](#)



Bhaskar Soni

Phase 10: Attacking Input Validations [Cont.]

Introduction to SQL vs NoSQL: [\[Link\]](#)

Abusing NoSQL databases: [\[Link\]](#)

Making cry - attacking NoSQL for pentesters: [\[Link\]](#)

Xpath and XML injection:

Introduction to Xpath injection: [\[Link\]](#)

Introduction to XML injection: [\[Link\]](#)

Practical 1 - bWAPP: [\[Link\]](#)

Practical 2 - Mutillidae: [\[Link\]](#)

Practical 3 - webgoat: [\[Link\]](#)

Hack admin panel using Xpath injection: [\[Link\]](#)

XXE demo 1: [\[Link\]](#)

XXE demo 2: [\[Link\]](#)

XXE demo 3: [\[Link\]](#)



Bhaskar Soni

Phase 10: Attacking Input Validations [Cont.]

LDAP injection:

Introduction and practical 1: [\[Link\]](#)

Practical 2: [\[Link\]](#)

OS command injection:

OS command injection in bWAPP: [\[Link\]](#)

bWAAP- OS command injection with Commiux: [\[Link\]](#)

Local file inclusion:

Detailed introduction: [\[Link\]](#)

LFI demo 1: [\[Link\]](#)

LFI demo 2: [\[Link\]](#)

Remote file inclusion:

Detailed introduction: [\[Link\]](#)



Bhaskar Soni

Phase 10: Attacking Input Validations [Cont.]

RFI demo 1: [\[Link\]](#)

RFI introduction and demo 2: [\[Link\]](#)

HTTP splitting/smuggling:

Detailed introduction: [\[Link\]](#)

Demo 1: [\[Link\]](#)



Bhaskar Soni

Phase 11 – Generating and testing error codes

Generating normal error codes by visiting files that may not exist on the server - for example visit `chintan.php` or `chintan.aspx` file on any website and it may redirect you to `404.php` or `404.aspx` or their customer error page. Check if an error page is generated by default web server or application framework or a custom page is displayed which does not display any sensitive information. Use BurpSuite fuzzing techniques to generate stack trace error codes: [\[Link\]](#)



Phase 12 – Weak cryptography testing

SSL/TLS weak configuration explained: [\[Link\]](#)

Testing weak SSL/TLS ciphers: [\[Link\]](#)

Test SSL/TLS security with Qualys guard: [\[Link\]](#)

Sensitive information sent via unencrypted channels: [\[Link\]](#)



Bhaskar Soni

Phase 13 – Business logic vulnerability

What is a business logic flaw: [\[Link\]](#)

The Difficulties Finding Business Logic Vulnerabilities with Traditional Security Tools: [\[Link\]](#)

How To Identify Business Logic Flaws: [\[Link\]](#)

Business Logic Flaws: Attacker Mindset: [\[Link\]](#)

Business Logic Flaws: Dos Attack On Resource: [\[Link\]](#)

Business Logic Flaws: Abuse Cases: Information Disclosure: [\[Link\]](#)

Business Logic Flaws: Abuse Cases: iPod Repairman Dupes Apple: [\[Link\]](#)

Business Logic Flaws: Abuse Cases: Online Auction: [\[Link\]](#)

Business Logic Flaws: How To Navigate Code Using ShiftLeft Ocular: [\[Link\]](#)



Bhaskar Soni

Phase 13 – Business logic vulnerability [Cont.]

Business Logic Security Checks: Data Privacy Compliance: [\[Link\]](#)

Business Logic Security Checks: Encryption Compliance: [\[Link\]](#)

Business Logic Security: Enforcement Checks: [\[Link\]](#)

Business Logic Exploits: SQL Injection: [\[Link\]](#)

Business Logic Exploits: Security Misconfiguration: [\[Link\]](#)

Business Logic Exploits: Data Leakage: [\[Link\]](#)

Demo 1: [\[Link\]](#)

Demo 2: [\[Link\]](#)

Demo 3: [\[Link\]](#)

Demo 4: [\[Link\]](#)

Demo 5: [\[Link\]](#)

Demo 6: [\[Link\]](#)



Bhaskar Soni

Enjoy Learning!
Thank You 😊

