Contents

Preface i

About the Author v

Co-Authors vi

About the Reviewers vii

Acknowledgments viii

I. Web Application Security 1

1. Web Application Architecture 2

1.1. Introduction to Web Application Penetration Testing 2
1.2. Introduction to HTTP 2
1.3. Web Application Architecture 7
1.4. Introduction to OWASP 9
1.4.1. Importance of OWASP in Modern Penetration Testing . . . 10
1.4.2. OWASP Top 10 Vulnerabilities 2021 11

2. Lab Setup 13

2.1. Why Lab Setup to Perform Penetration Testing 13
2.2. Overview of the Lab Environment: 13
2.2.1. BookShelf Vulnerable Web Application 13
2.2.2. XVWA (Xtreme Vulnerable Web Application) 14
2.3. Setting up the lab 15
2.3.1. Download the Required VMs 15
2.3.2. Installing VirtualBox 15
2.3.3. Importing the BookShelf Application using the OVA File . 16
2.3.4. Installing Kali Linux and XVWA Using ISO Files 16
2.3.5. Adjust VM Settings for Better Performance 17
2.3.6. Enabling Shared Clipboard and Drag and Drop 17
2.3.7. Configuring Network for Testing 18
2.3.8. Verifying Network Connectivity 18
2.3.9. Accessing the Vulnerable Web Applications 19

3. SQL Injection (SQLi) 21

3.1. Introduction to SQL Injection 21
3.1.1. How SQL Injection Works 21
3.1.2. Types of SQL Injection 22

3.1.3. Real-World SQL Injection Attacks 23
3.2. From Theory to Practice: SQL Injection Exploitation 24
3.2.1. Authentication Bypass using SQL Injection 24
3.2.2. Error-Based SQL Injection 26
3.2.3. Blind SQL Injection 31
3.2.4. Automating SQL Injection Attacks with sqlmap 34
3.3. Preventing SQL Injection 37

4. Cross Site Scripting (XSS) 39
4.1. Introduction 39
4.2. Introduction to Same Origin Policy 39
4.2.1. Origin 39
4.2.2. Cross-Origin Interactions 40
4.2.3. Same Origin Policy (SOP) 40
4.3. Introduction to Cross Site Scripting (XSS) 40
4.4. Types of Cross Site Scripting (XSS) 41
4.5. Testing for XSS Vulnerabilities 43
4.5.1. Testing for Reflected XSS 43
4.5.2. Testing for Stored XSS 44
4.5.3. Testing for DOM-Based XSS 45
4.6. XSS in BookShelf Application 46
4.6.1. XSS BookShelf Application via SVG uploads 47
4.6.2. Exploiting XSS - Cookie Stealing 48
4.7. Preventing XSS 50

5. Cross Site Request Forgery 52
5.1. Introduction to Cross-Site Request Forgery (CSRF) 52
5.1.1. Conditions Required for Cross-Site Request Forgery (CSRF) 52
5.2. Understanding CSRF with an example 53
5.3. CSRF demonstration using XVWA 54
5.4. Preventing Cross-Site Request Forgery (CSRF) 60

6. XML Vulnerabilities 62
6.1. XPath Injection 62
6.1.1. What is XPath Injection? 62
6.1.2. XPATH Injection demonstration using BookShelf 62
6.1.3. Preventing XPath Injection 63
6.2. External XML Entity (XXE) Injection 64
6.2.1. What is an Entity? 64
6.2.2. Types of Entities 65
6.2.3. Exploiting XXE (External XML Entity) 66
6.2.4. XXE testing tips for security practitioners 68

6.2.5. Introduction to Blind XXE 69
6.2.6. Exploiting Blind XXE for data exfiltration 71
6.3. Preventing XXE 76
6.3.1. Disabling external entity parsing 76
6.3.2. Validating input when entities are required 76
6.3.3. Using Java's Secure Processing Features 76

7. Access Control Issues 78
7.1. Introduction to Access Control in Web Applications 78
7.1.1. Authentication: Establishing Identity 78
7.1.2. Authorization: Defining Access 79
7.1.3. The Connection Between Authentication and Authorization 79
7.1.4. Weaknesses in Authorization 79
7.2. Privilege Escalation in the Bookshelf application 80
7.3. Implementing Role Based Access Controls 81

8. File Upload Vulnerabilities 84
8.1. Introduction to File Uploads 84
8.2. Abusing File Uploads 84
8.3. Preventing File Upload Vulnerabilities 88

9. Platform Misconfigurations 90
9.1. Introduction 90
9.2. Exploiting CVE-2017-5638 91
9.3. Exploiting Misconfigured Tomcat 93
9.4. Mitigations and best practices 94

10. Improper Error Handling 97
10.1. Introduction to improper error handling 97
10.2. Demonstrating improper error handling in BookShelf 98
10.3. Preventing Improper Error Handling 101

11. Java Deserialization 103
11.1. Introduction to Java Deserialization 103
11.2. Understanding Serialization and Deserialization in Java 103
11.2.1. The Serialization Process 103
11.2.2. The Deserialization Process 104
11.2.3. Important Characteristics of Serialization 106
11.2.4. Security Implications 107
11.3. Practical Demonstration of Deserialization Attacks 107
11.3.1. Black Box Identification of Java Serialization 109
11.3.2. White Box Identification of Java Serialization 112
11.3.3. Stored XSS via Insecure Deserialization 115

11.3.4. Exploiting Java Deserialization using Ysoserial 119
11.3.5. Java Deserialization Proof-of-Concept (PoC) Using URLDNS 123
11.4. Java Deserialization Prevention 127

12. Broken Cryptography 130
12.1. Cryptography 130
12.1.1. Types of Cryptography 130
12.1.2. Objectives of Cryptography 131
12.2. The CIA Triad in Information Security 132
12.3. Encoding vs Encryption vs Hashing 133
12.4. The Use of Cryptography in Applications 134
12.4.1. Transport Layer Security (TLS) 135
12.4.2. Data at Rest 135
12.4.3. End-to-End Encryption (E2EE) 136
12.4.4. Frameworks 137
12.4.5. Customized Application Features 138
12.5. Password Cracking with Hash Identifier and John the Ripper . . . 138
12.6. Improper Crypto Usage – Case Study: Linux.Encoder 140
12.7. Post-Quantum Cryptography (PQC) 143
12.7.1. Quantum Computers and Qubits 143
12.7.2. Public-Key Cryptography and Quantum Threats 144
12.7.3. Symmetric Cryptography and Grover's Algorithm 145
12.7.4. Harvest Now, Decrypt Later (HNDL) 146
12.7.5. State of Quantum Hardware 147
12.7.6. Post-Quantum Cryptography (PQC) and NIST Standards . 147
12.7.7. Crypto Agility 150
12.7.8. Summary 151

13. Automated Vulnerability Scanning 152
13.1. Introduction 152
13.2. Unauthenticated Scans using ZAP 152
13.3. Authenticated Scans Using ZAP 155
13.4. Can Automated Scanners Discover all Vulnerabilities? 157

14. Web Security in the age of Artificial Intelligence (AI) 158
14.1. Introduction 158
14.2. How LLMs work in the modern web stack 159
14.3. The Risks of AI-Powered Applications 159
14.3.1. OWASP Top 10 for Large Language Model Applications (v1.1) 160
14.3.2. MITRE ATLAS - Understanding AI specific attack patterns 161

14.4. Introducing BookShelf AI 163
14.4.1. Application Architecture 163
14.4.2. Embedded Vulnerability Categories 164
14.5. Lab Setup and Configuration 164
14.5.1. Installing Prerequisites 164
14.5.2. Project Setup 165
14.5.3. Exploring BookShelf AI Application 167
14.6. Exploiting BookShelf AI vulnerabilities 171
14.6.1. Prompt Injection 172
14.6.2. Exploiting Contextual Database Access 177
14.6.3. SQL Injection Vulnerabilities 183
14.7. Best Practices for AI-Powered Applications 186

II. JSON Web Token (JWT) Security 187

15. The JWT Security Landscape 188
15.1. The Evolution of Authentication: From Passwords to Tokens . . . 188
15.2. The Rise of Token-Based Authentication 188
15.3. The JWT Security Paradox 189
15.4. Real-World Impact 189
15.5. Approaching This Guide 190

16. Setting Up JWT Attack Lab 191
16.1. Virtual Environment Setup 191
16.2. Installing Postman 193
16.3. API Documentation - "Secure Store" 194
16.4. Wrapping Up 196

17. Introduction to JSON Web Tokens 197
17.1. What is JWT ? 197
17.2. JWT Structure 197
17.3. How JWT Works? 200
17.4. JWT Demonstration 201

18. JWT In Depth 204
18.1. Common JWT Algorithms 204
18.2. HS256 and it's application in JWT 204
18.2.1. What is HS256? 204
18.2.2. How HS256 Works 205
18.2.3. HS256 in Action 205
18.3. RS256 and it's application in JWT 208
18.3.1. What is RS256? 208

18.3.2. How RS256 Works 208
18.3.3. RS256 in Action 209
18.4. None Signature 212

19. JWT Command Line Kung-Fu 214

19.1. Overview 214
19.2. Manually Crafting an HS256 Signature 214
19.3. Generating RSA Keys Using OpenSSL 217
19.4. Manually Creating RS256 Signature 219
19.5. Summary 221

20. JWT Attacks and Defenses 222

20.1. Abusing the None Algorithm 222
20.2. Signature Stripping Attack 225
20.3. Cracking Weak HS256 Secrets 227
20.4. Substitution Attack 229
20.5. Defenses 232

21. Final Thoughts 234

22. References 235