L. M. Saxton NS_2023

Vulnerability Audit and Assessment of

https://bookacheckup.co.uk/index.php

Baseline Analysis and Plan

Table of Contents

1. Introduction

The following audit serves as a plan for the penetration testing of *http://bookacheckup.co.uk/index.php*, a healthcare appointment-booking website. This audit will examine and suggest:

- healthcare service and privacy regulations
- PHP web application vulnerabilities
- an application-specific attack surface
- tools for vulnerability detection
- a complete penetration test timeline

Testing results should provide guidance for subsequent cyber-risk management.

2. Security Statutes and Application Vulnerability

2.1 Healthcare Information Privacy

The following regulations are relevant to both healthcare service quality and patient privacy:

- GDPR (2019)
    - security of patient appointment records
    - security of patient health history
- HIPAA (1996)
    - security of financial and personal information
    - security of public health information and insurance records

- NIST (Barker et al., 2009)
    - reliable service accessability

These standards have informed the following penetration testing foci.

2.2. PHP Vulnerabilities

PHP-based applications have demonstrated attack vulnerabilities (*Table 1*; see *Appendix I*). The below have historical precident and should be thoroughly vetted during testing.

Table 1: Common PHP Vulnerabilities

Attack Name	**Attack Type** (Mitre, 2023)	**Possible Attack Vector**	**Source**
Attacker-Controlled Input	Data modification	Application layer	Edmunds, 2016
Brute Forcing	Privilege elevation	Backend login	Mitre, 2023
Code Injection	Data modification	Input field	Backes et al., 2017
Modifying Cookies	Data modification	Application layer	Mitre, 2023
Cross-site Request Forgery	Privilege elevation	Backend login	Pinto & Stuttard, 2011
Cross-site Scripting	Unauthorized command execution	Input field	Gupta & Gupta, 2015
Denial of Service	Service disruption	Application layer	Shimatikov & Son, 2011
File Inclusion	Unauthorized command execution	Input field	Gong & Zhao, 2015
Missing Authorization Checks	Privilege elevation	Backend login	Shimatikov & Son, 2011
SQL Injection	Unauthorized command execution	Input field	Backes et al., 2017

3. Penetration Testing

Though penetration tests are an essential aspect of application security management, limitations (Pinto & Stuttard, 2011) such as

- undetectable vulnerabilities
- inaccuate attack severity
- penetration tester skill

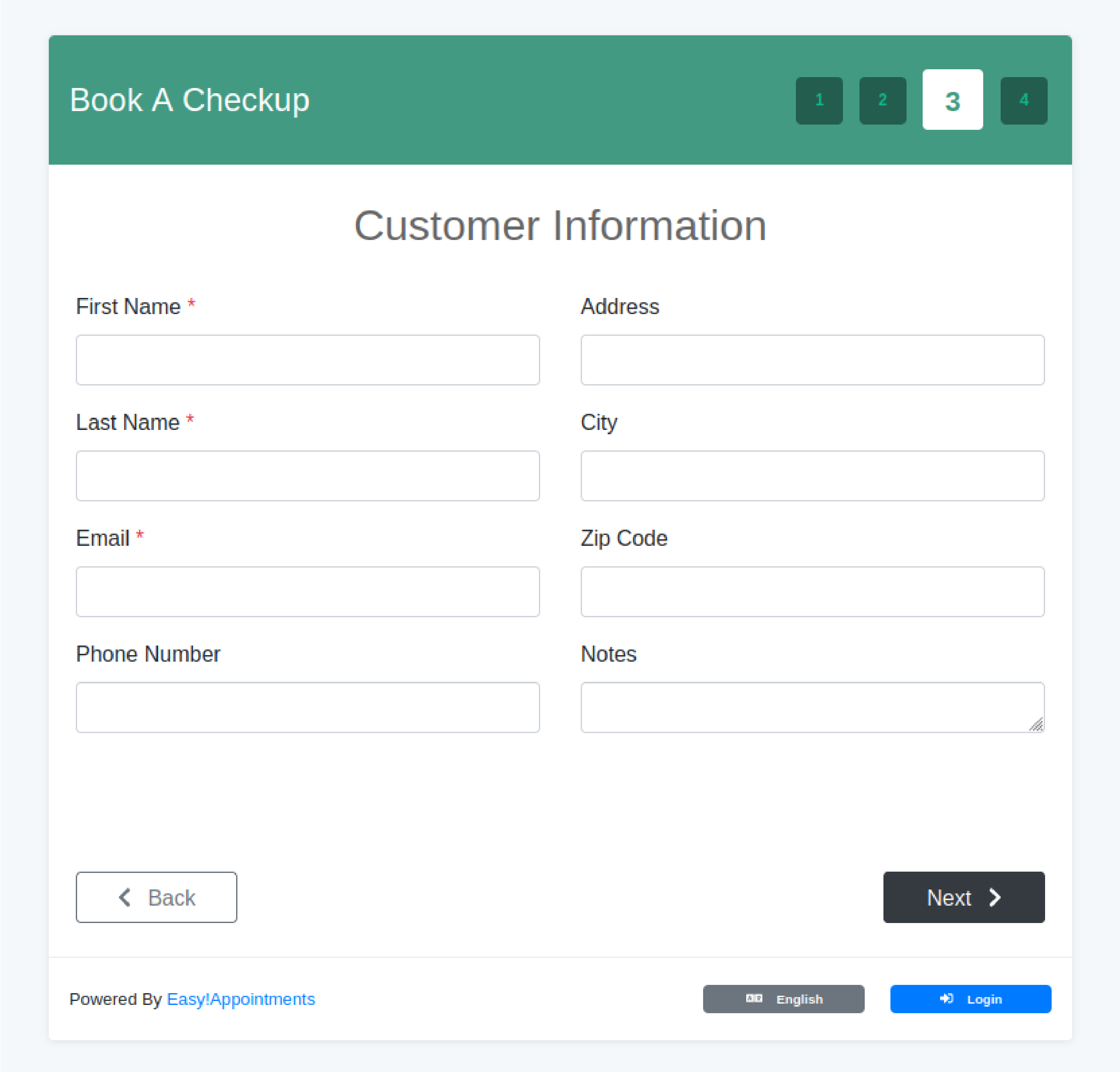
may impact test findings. These limitations should be recognized and further action taken, if necessary.

3.1 Application-Specific Testing

Prelimiary reconaissance of *http://bookacheckup.co.uk/index.php* found the following relevant attack surfaces:

- HTTP (application layer)
    - attacker-controlled input, cookie modification, DOS attack
- Customer information page (*Figure 1*)
    - injection attacks, file inclusion, cross-site scripting
- Backend section (*Figure 2*)
    - brute forcing, forgery attacks, authorization attacks, injection attacks, cross-site scripting
- Password regeneration page (*Figure 3*)
    - forgery attacks, scripting attacks, injection attacks, cross-site scripting

Actual testing would encompass any newly discovered attack surfaces. Any DOS or brute force testing is recommended outside normal operation hours to avoid service disruption.


Book A Checkup

Customer Information

First Name *

Last Name *

Email *

Phone Number

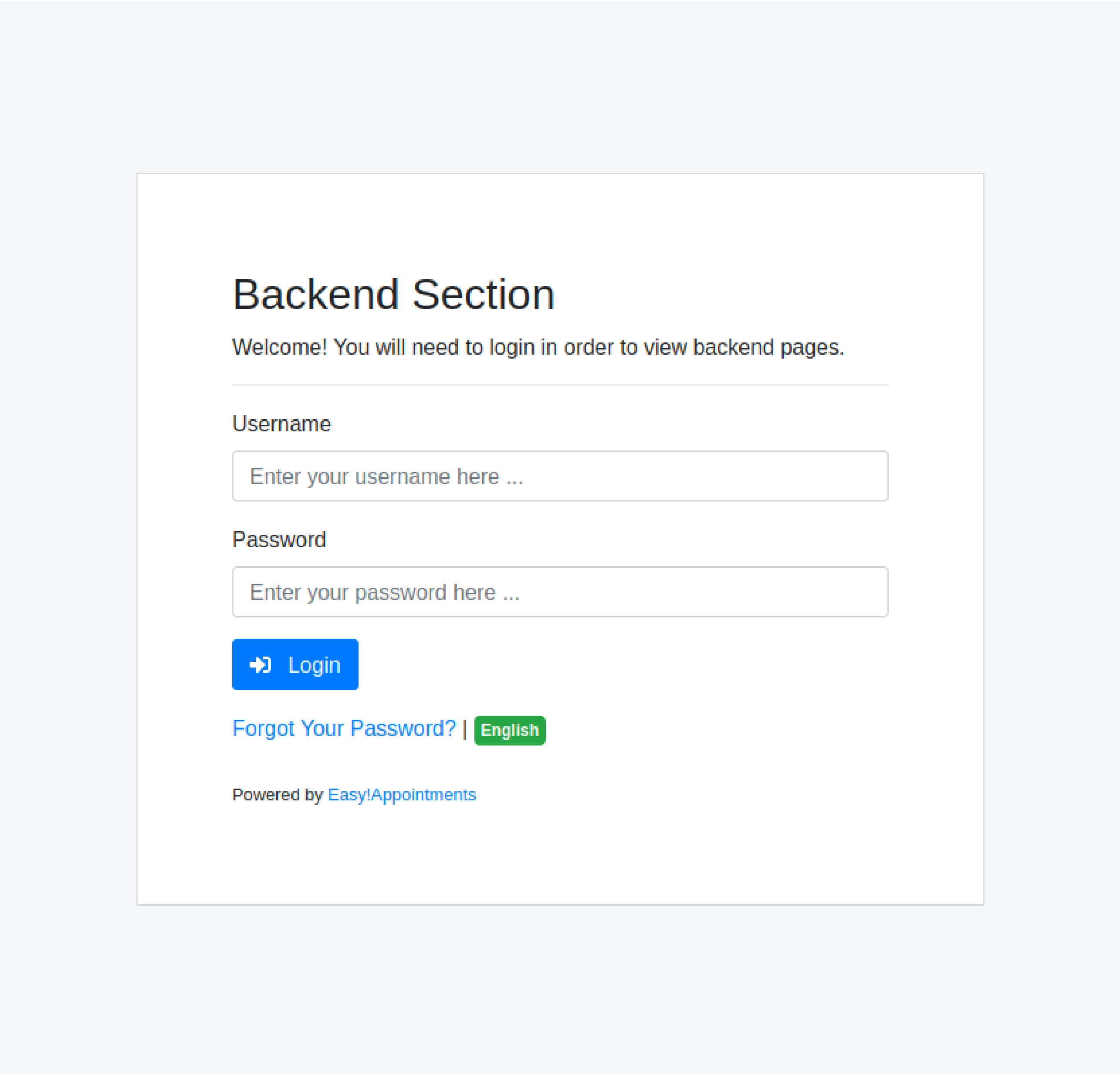
Address

City

Zip Code

Notes

Figure 1: Customer Information



Backend Section

Welcome! You will need to login in order to view backend pages.

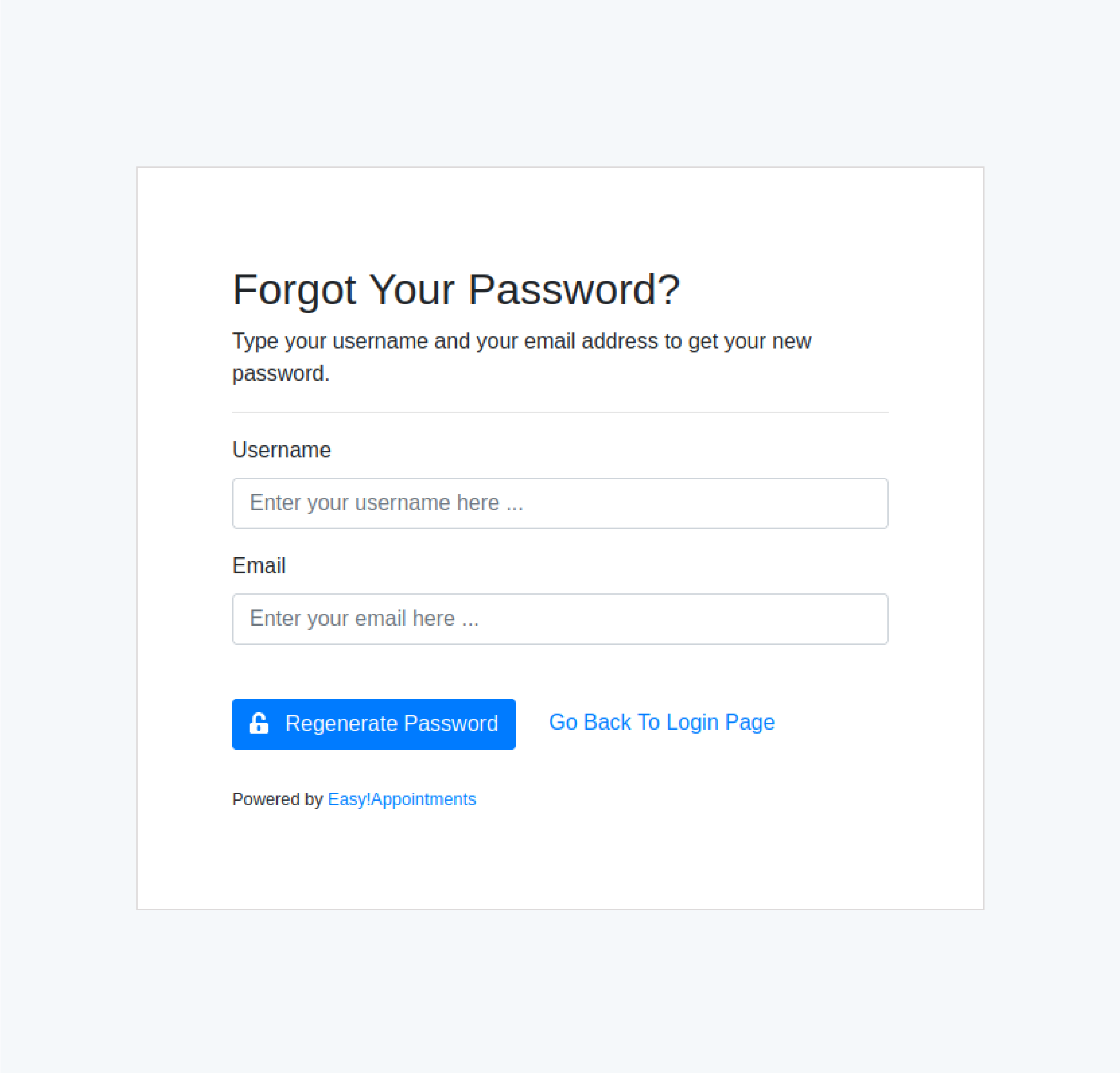
Username

Enter your username here ...

Password

Enter your password here ...

Figure 2: Backend Section



Forgot Your Password?

Type your username and your email address to get your new password.

Username

Enter your username here ...

Email

Enter your email here ...

Regenerate Password

Go Back To Login Page

Powered by Easy!Appointments

Figure 3: Password Regeneration

3.2. Penetration Testing Tools

The following tools would be utilized during the penetration test:

- Nmap (Aharoni et al., 2011; Kaur & Kaur, 2017)
  - port scanning
    - benefits: many tool options, integrates well with other suites
    - limitations: IP/Network scan only

- Burpsuite (Khawaja, 2018; Li, 2021)
  - proxy server
    - benefits: includes a spider scanner, intruder tool, and repeater tab
    - limitations: some functions are license-only

- Nessus (Chauhan, 2018; Pauli, 2013)
  - vulnerability scanning
    - benefits: wide vulnerability range on networks and hosts
    - limitation: community version plug-ins are behind professional versions

- Metaploit (Aharoni et al., 2011; Jaswal & Rahalkar, 2019)
  - vulnerability scanning and exploitation
    - benefits: SMB Logins scan and third-party scan exploitation
    - limitations: better suited for infrastructure exploitation than web applications

4. Penetration Testing Timeline

Based on industry standards (Majiah, 2017), a penetration test for the vulnerabilities above would require 17 days and 4.5 hours (Table 2).

Table 2: Penetration Testing Timeline

Activities	Tools	Duration
Web Application Reconaissance	Search engine	1 day
Network Scan	Nmap	2 hours
Fingerprint Web Application	Burpsuite	1 day
Attacker-Controlled Input	Burpsuite	2 hours
Brute Forcing	Metasploit	2 days
Code Injection	Nessus	1 day
Cookie Modification	Burpsuite	30 minutes
Cross-site Request Forgery	Nessus	1 day
Cross-site Scripting	Burpsuite	2 days
Denial of Service	Nessus	1 day
File Inclusion	Burpsuite	1 day
Missing Authorization Checks	Burpsuite	1 day
SQL Injection	Nessus	2 days

Inclusive of the final report, findings would be presented 19 days after penetration test commencement.

5. Conclusion

In this audit healthcare and cyber privacy statues, application attack surface, penetration testing tools, and a timeline for testing have been proposed and discussed. Results are meant to guide subsequent cyber-risk management.

6. Appendices

6.1 Appendix I

The attack classifications below have been assembled under the CAPEC framework (Mitre, 2023).

Attack Name	Attack Likelihood	Attack Severity	Skill Level Required	
Attacker-Controlled Input	Medium	Medium	n/a	
Brute Forcing	n/a	High	Low	
Code Injection	High	High	n/a	
Modifying Cookies	High	High	Low	High
Cross-site Request Forgery	High	Very high	Medium	
Cross-site Scripting	High	Very high	Low	High
Denial of Service	High	Medium	n/a	
File Inclusion	High	High	Low	Medium
Missing Authorization Checks	High	Medium	Low	
SQL Injection	High	High	Low	

7. References

Aharoni, M., Kearns, D., Kennedy, D., & O’Gorman, F. (2011) *Metasploit: The Penetration Tester's Guide*. San Francisco: No Starch Press.

Backes, M.Rieck, K., Skoruppa, M., Stock, B., & Yamaguchi, F. (2017) “2017 IEEE European Symposium on Security and Privacy,” in *Efficient and Flexible Discovery of PHP Application Vulnerabilites*. Paris, FR: IEEE: 334–349.

Barker, E. Branstad, D., Chokani, S., & Smid, M. (2009) *Cryptographic Key Management Workshop Summary*. tech. Gaithersburg, MD: NIST: 1–59.

Chauhan, A.S. (2018) *Practical Network Scanning: Capture Network Vulnerabilities Using Standard Tools Such as Nmap and Nessus*. Mumbai, IN: PACKT Publishing Limited.

Edmunds, B. (2016) *Securing PHP Apps*. Berkeley, USA: Apress.

Gong, R. & Zhao, J. (2015) “A New Framework of Security Vulnerabilities Detection in PHP Web Application,” *2015 9th International Conference on Innovative Mobile and Internet Services in Ubiquitous Computing:* 271–276. Available at: https://doi.org/10.1109/imis.2015.42.

Gupta, B.B. & Gupta, S. (2015) “PHP-Sensor: A Prototype Method to Discover Workflow Violation and XSS Vulnerabilities in PHP Web Applications,” *Proceedings of the 12th ACM International Conference on Computing Frontiers:* 1–8. Available at: https://doi.org/10.1145/2742854.2745719.

Health Insurance Portability and Accountability Act of 1996 (1996) *govinfo.gov*. Government of the United States of America. Available at: https://www.govinfo.gov/content/pkg/PLAW-104publ191/pdf/PLAW-104publ191.pdf (Accessed: February 13, 2023).

Jaswal, N. & Rahalkar, S. (2019) *The Complete Metasploit Guide: Explore Effective Penetration Testing Techniques with Metasploit*. Birmingham, UK: Packt Publishing.

Kaur, G. & Kaur, N. (2017) “Penetration Testing – Reconnaissance with NMAP Tool,” *International Journal of Advanced Research in Computer Science*, 8(3): 844–846.

Khawaja, G. (2018) *Practical Web Penetration Testing: Secure Web Applications Using BURP Suite, Nmap, Metasploit, and More*. Birmingham, UK: Packt Publishing.

Li, V. (2021) Bug Bounty Bootcamp: The Guide to Finding and Reporting Web Vulnerabilities. San Francisco, USA: No Starch Press.

Majiah, A.E. (2017) *How Long Does It Take to Do a Penetration Testing?*, *LinkedIn*. Available at: https://www.linkedin.com/pulse/how-long-does-take-do-penetration-testing-aldo-elam-majiah (Accessed: February 13, 2023).

Mitre. Common Attack Pattern Enumeration and Classification (2023) CAPEC. Available at: https://capec.mitre.org/data/definitions/658.html (Accessed: February 13, 2023).

Pinto, M. & Stuttard, D. (2011) The Web Application Hacker’s Handbook: Finding and Exploiting Security Flaws. 2nd Ed. Indianapolis, USA: Wiley.

Pauli, J. (2013) *The Basics of Web Hacking*. Boston, USA: Elsevier Science.

Recital 35 - Health Data (2019) *General Data Protection Regulation (GDPR)*. Available at: https://gdpr-info.eu/recitals/no-35/ (Accessed: February 13, 2023).

Son, S. & Shmatikov, V. (2011) “SAFERPHP: Finding Semantic Vulnerabilities in PHP Applications,” *Proceedings of the ACM SIGPLAN 6th Workshop on Programming Languages and Analysis for Security:* 1–13. Available at: https://doi.org/10.1145/2166956.2166964.