L. M. Saxton NS_2023

Assignment 1:

1. Introduction (50 words)

- Importance of web app security, as it can lead to attacks in lower IP/TCP layers
  - the goal of this summary is to present a blueprint for locating vulnerabilities that leave the app open to breach
- What will be discussed in this report
  - privacy concerns for health information (GDPR & HIPAA)
  - most relevant vulnerabilities in healthcare
  - most common vulnerabilities for php websites
  - Tools which can scan for and detect these vulnerabilities for subsequent mitigation

(50 words)

2. Relevant Vulnerabilities (100 words)

2.1 Privacy Concerns for Healthcare

- HIPAA concerns (HIPAA, xxxx)
  - data privacy – Access controls, Authentication (Gauthier & Merlo, 2012, web app handbook)
  - access to service
- GDPR concerns (GDPR, 2018)
  - privacy concerns
  - human element (Human engineering book)

2.2 Demonstrated PHP Vulnerabilities

- Table with the following layout:
  - Attack vulnerability
  - Type of attack
  - Area of website which is vulnerable
  - Source
- Discuss the different attack types (with an image showing example of each type) according to Mitre framework.
  - DOS “due to attack-controlled infinite loop” (Shimatikov & Son, 2011:2
  - Missing authorization checks (ibid)
  - Cross-site Scripting (Gupta & Gupta, 2010)
  - Workflow violation (ibid)
  - File Inclusion (Gong & Zhao, 2015)
  - SQL Injection (Backes et al., 2017)
  - Command Injection (ibid)
  - Code Injection (ibid)
  - Attacker-Controlled Input (PHP book)
  - CSRF (Web app handbook)
  - Password/username bruteforcing (xxxx)
  - SSL certificate (xxxx)

(150 words)

3. Penetration Testing (50 words)

- Benefits of Pentesting (pentesting book)
  - Why should companies pentest? (pentesting articles)
  - What happens if companies do not pentest? (pentesting articles)
- Limitations of Pentesting (web app handbook)

   - ○ Scans are cursory – “Like knowing a window can be broken by a stone but not throwing the stone” – cannot truly assess the impact, though Mitre and others should be utilized.
   - ○ Cannot find all vulnerabilities, only some
   - ○ Is only as good as the pentester

(200 words)

3.1 Pentesting Specification for xxx.php (150 words)

- In order to provide the most releveant pentest, the following assumptions have been made:
- ○ the pentest will focus on the vulnerabilities which can be accessed at the application layer of the network (add attack surface table here)
- ■ The pentest itself will be a black-box test (bug bounty hunting) = remote and dynamic to better mirror an actual web app attack
- ■ any attack surface/vulnerability scans will be manual so as to lessen any unintentional denial of service (add tool table here)
- ■ Password brute force is recommended but may cause denial of service
- ■ only the information/forms provided on the website will be utilized
- ■ Vulnerabilities within the application layer will be documented and exploited
- ■ Vulnerabilities outside the application layer will be documented but not exploited
- ■ Any possibility of denial or service will be documented but not exploited
- ○ The following attack surface is relevant to the pentest based on a prelimiary scan (table – area, relevant attack, source)
  - ■ user form fields
  - ■ hidden form fields
  - ■ server side attacks
  - ■ client-side attacks
  - ■ human engineering
- ○ Tools to use (Table – tool, relevant attacks, source)
  - ■ tool 1
  - ■ tool 2
  - ■ etc

3.1.1 Denial of Service Probability (50 words)

- Not the intention of the pentester to cause a denial of service during testing, however the chance of disruption during scanning does exist. May be prudent to perform attack surface and vulnerability scans during off-hours to reduce chance of service disruption in line with HIPAA/GDPR

(400 words)

3.2 Pentesting Timeline (100)

Depends on:

given the size and scope of xxxx.php the following timeline seems appropriate to perform a pentest:

(change this to an hour by hour/attack specific with linkedin article)

1. Day 1 – cursory fingerprinting – pentester will manual go through the website taking note of the
2. Day 2 – attack surface scan – using X tool, the website will be scanned for attack vectors
3. vulnerability scan – Using tools x, y, z attack vectors will be scanned for exploitable vulnerabilities
4. Day 4-6 – vulnerability documentation – based on the previous scans, an attack framework will be compiled and assessed

5. Day 7-8 – Report compilation – based on the vulnerability documentation a detailed executive summary of the vulnerability findings will be presented
6. Day 9 – Report delivery – The report will be presented and discussed with relevant parties

Should this be an image or a table?
(500 words)

4. Conclusion (100 words)
(600 words)

Attack Name	Attack Type	Possible Attack Vector	Source
Denial of Service (DOS)			Shimatikov & Son, 2011
Missing Authorization Checks			Shimatikov & Son, 2011
Cross-site Scripting (XSS)			Gupta & Gupta, 2010
Workflow Violation			Gupta & Gupta, 2010
File Inclusion			Gong & Zhao, 2015
SQL Injection			Backes et al., 2017
Command Injection			Backes et al., 2017
Code Injection			Backes et al., 2017
Attacker-Controlled Input			PHP Book, xxxx
Cross-site Request Forgery (CSRF)			Web app handbook, xxxx
Cookie Tampering			Mitre, 2023
Brute Forcing			Mitre, 2023

Appendix I

<table>
<tr><th>Attack Name</th><th>Attack Likelihood</th><th>Attack Severity</th><th colspan="2">Skill Level Required</th></tr>
<tr><td>Attacker-Controlled Input</td><td>Medium</td><td>Medium</td><td colspan="2">n/a</td></tr>
<tr><td>Brute Forcing</td><td>n/a</td><td>High</td><td colspan="2">Low</td></tr>
<tr><td>Code Injection</td><td>High</td><td>High</td><td colspan="2">n/a</td></tr>
<tr><td>Modifying Cookies</td><td>High</td><td>High</td><td>Low</td><td>High</td></tr>
<tr><td>Cross-site Request Forgery</td><td>High</td><td>Very high</td><td colspan="2">Medium</td></tr>
</table>

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	