|  |  |  |  |
| --- | --- | --- | --- |
| Module Code | DWD 507 | Module Title | Security and Testing |
| Student ID | 3201807 | Name | Jack Bradley-Coombes |

***Antique Mall* requirements:**

1. Using appropriate tools or techniques perform security testings for the *Antique Mall* website <http://antique-mall.507testdwdd.dipwww.visioncollege.ac.nz/> and document the process in a format as below. [22 marks]

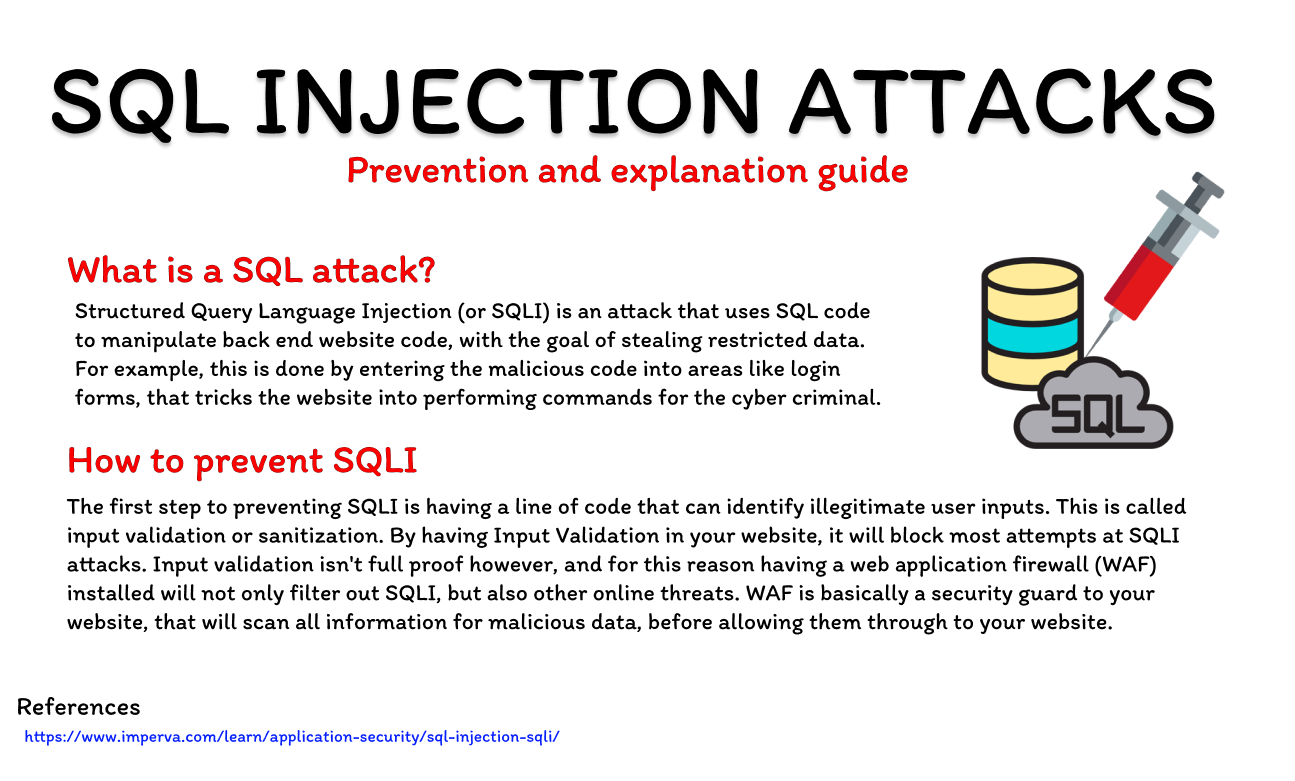
*Note: I used a locally ran ghost server for this assignment*

|  |  |
| --- | --- |
| **a.** | **Vulnerability Scanning** [8 marks] |
| Tool/Technique used | ZAP |
| Purpose | To test the websites for security vulnerabilities in the interest of securing the website. |
| Testing result  (summary) | 12 minor vulnerabilities were found, including some updated library’s, unnecessary files, and unimplemented protection Methods. |
| Recommendations | **Error – Content Security Policy Header Not Set:** Ensure that the web server, app server, and load banner is configured to set the Content-Security-Policy header. This is an added layer of security that helps detect XSS and Data injection attacks.  **Vulnerable JS Library:** The Current version of the library “ember” is vulnerable, recommended to update to the latest version.  **Missing Anti-Clickjacking header:** There is no protection against “Clickjacking”. Ensuring Content-Security-Policy or X-Frame-Option HTTP Header is set on all web pages, will remedy this issue.  **Absence of Anti-CSRF Tokens:** Lack of Anti-CSRF tokens in the HTML submission form leaves it vulnerable to CSRF attacks. Using a vetted library or framework design, like an anti-CRSF package, will help protect this weakness. |
| Evidence of the testing |  |
| **b.** | **Registration/Login Form- Use of strong password and validation** [4 marks] |
| Tool/Technique used | Manual test. *(The application for password testing would not work)* |
| Purpose | To test if the login form checks for valid information and strong passwords. |
| Testing result  (summary) | There is no password section to test, but email validation works. |
| Recommendations | Adding a password requirement for accounts, with use of strong password validation would add security to user accounts. |
| Evidence of the testing |  |
| **c.** | **URL Parameter Tampering attack to access privileged user accounts** [3 marks] |
| Tool/Technique used | ManualTesting (Inspect Element) |
| Purpose | Testing that the login screen can’t be bypassed. |
| Testing result(summary) | Using the inspect element to find URLs to bypass the login screen resulted in leading to an error page. |
| Recommendations | If this error pages that these URLs lead to are error pages because there is no page made yet past the login screen, then once a page has been made ensuring it is encrypted and firewalled is a good idea. |
| Evidence of the testing |  |
| **d.** | **Malware Scanning** [3 marks] |
| Tool/Technique used | [**VirusTotal - Home**](https://www.virustotal.com/gui/home/url)*Note: I used the ‘antique mall’ website link for this test due to my ghost server being locally ran and not detected by the anti-malware sites.* |
| Purpose | To test website software is safe for end users. |
| Testing result (summary) | No files were marked as suspicious. (As website wasn’t public, I wasn’t able to scan it. I scanned the Antique mall website instead as directed by tutor) |
| Recommendations | Although no suspicious files were found, it’s a good idea have a good firewall and keep everything up to date. |
| Evidence of the testing |  |
| **e.** | **SQL injection** [4 marks] |
| Tool/Technique used | Manual injection into sign in form. |
| Purpose | To test for vulnerabilities to basic SQL injections. |
| Testing result  (summary) | The SQL injection attempt failed, the login form was valid and asked for a valid email instead. |
| Recommendations | Keeping all input validation and firewalls up to date on the website will help keep secure. |
| Evidence of the testing |  |

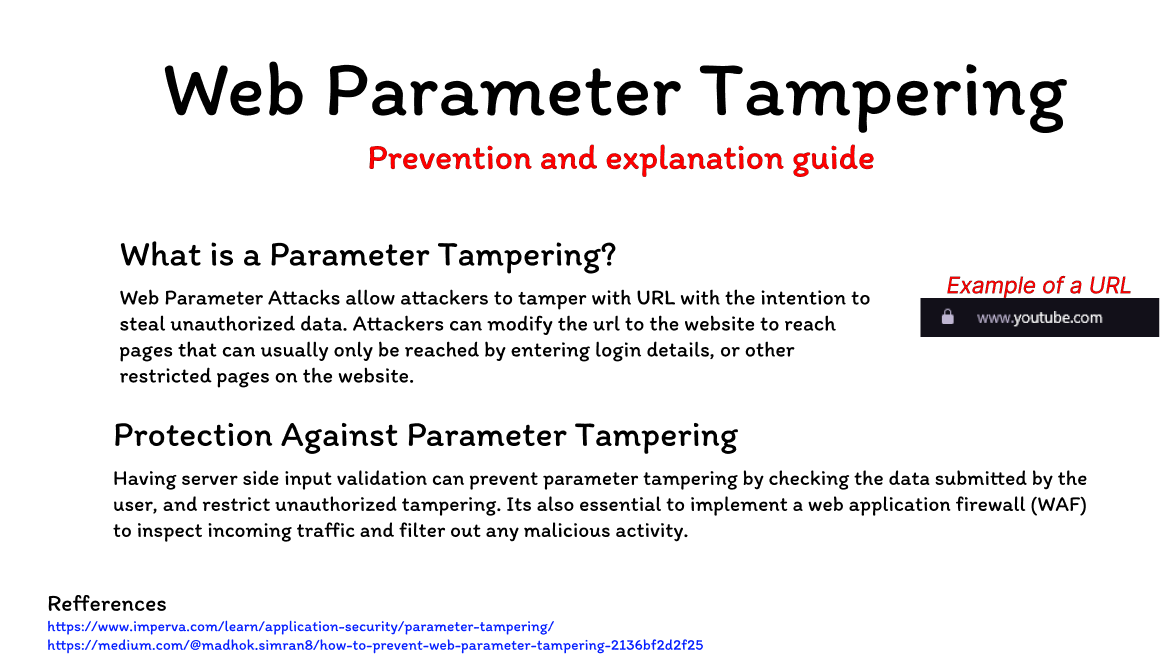
**NOTE:** You will need the testing result for *DWD 507- Assessment 2*.

1. Create cybersecurity user documentation, which should contain guidelines to prevent the top four possible cybercrimes applicable to *Antique Mall* business. The guidelines should provide sufficient information which is easy to understand for *Antique Mall* staff including, non-technical staff. You may use pictures, diagrams, charts and the like wherever applicable. [8 marks]

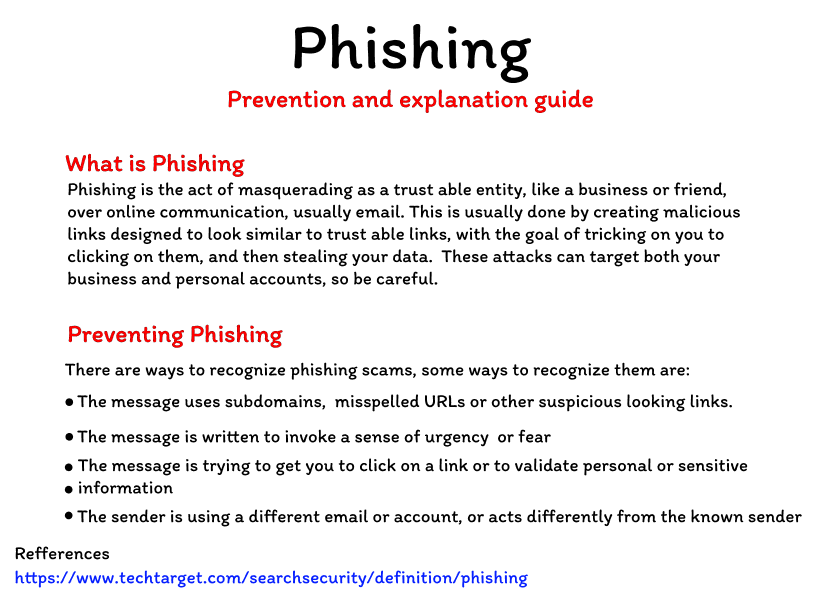
1: SQLI Attacks



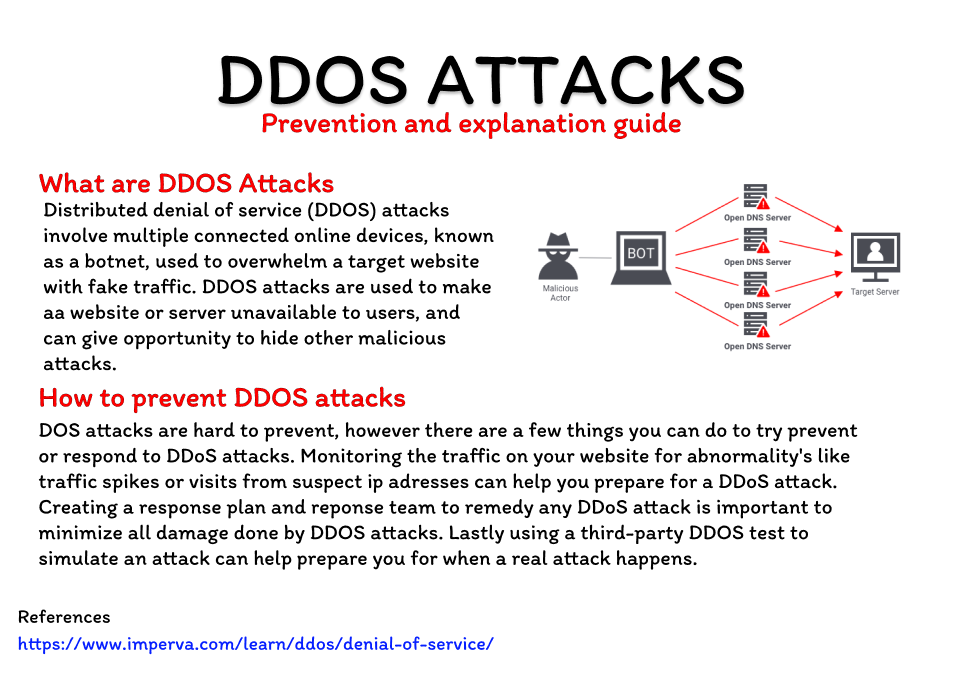
2: URL Tampering



3: Phishing



4: DDOS



References:

<https://www.imperva.com/learn/application-security/sql-injection-sqli/>

<https://medium.com/@madhok.simran8/how-to-prevent-web-parameter-tampering-2136bf2d2f25>

<https://www.imperva.com/learn/application-security/parameter-tampering/>

<https://www.techtarget.com/searchsecurity/definition/phishing>