**CS140 Computer Security**

**Coursework 2b: Looking for Security Problems in a Web Server**

**Answer form**

|  |
| --- |
| Name:  Manpreet Kaur Bahtra |
| Student ID: 2208321 |

Type your answers in the form. If you require extra space, please copy and paste more defect boxes.

Submit the form on Tabula by 12 noon on Monday, Jan 23, 2023. If you submit late without an extension having been granted, marks will be deducted according the standard University policy (5% per day or part thereof).

|  |  |  |
| --- | --- | --- |
| Describe the defect | In /httpd/conf httpd.conf file "AllowOverride" is set to all which allows the system to override Apache settings. If an intruder gets hold of the file system, they can modify the behaviour of the website as their commands will override apache settings. It overrides Apache setting for /var/www which contains files for the web server. | For markers' use only |
| How would you correct the problem? | Set AllowOverride to None (so that nothing can override Apache settings). |  |

|  |  |  |
| --- | --- | --- |
| Describe the defect | In /var/www/html/bank privateData.txt file password is visible, which means intruder can change the password since they know the current password. Intruder will also be able to block access to the intended user and will be withdraw funds from the bank account. Also, private details such as sortcode and account number are accessible. | For markers' use only |
| How would you correct the problem? | store password in a hashed format, e.g. using SHA 512. |  |

|  |  |  |
| --- | --- | --- |
| Describe the defect | In index.php file, there is no closing tag for body. | For markers' use only |
| How would you correct the problem? | add </body> |  |

|  |  |
| --- | --- |
| Describe the defect | In /var/www/html/others/CS140 temp1.txt file secure=false which means cookies can be communicated even if the connection is insecure. For example http instead of https. This means that attacker can steal cookie data and web browser will allow it since it allows insecure protocols. This is important because the cookie contains sensitive information such as username and password. Network Eavesdropping can occur in which attacker can use or modify the cookies to impersonate a user. |
| How would you correct the problem? | Set secure to true so that only https protocol is used to request. |

|  |  |  |
| --- | --- | --- |
| Describe the defect | In /var/www/html/bank login.php, after the name and password are inputted by the user, they are displayed in the web address bar as the input parameters. This is a breach of security as unwanted users can use those username and password and login. They can also change the password and restrict access to the intended user of their bank account. Also, the request method used is GET not POST. POST is secure and the contents of HTTP requests can be encrypted. It could also result in Phishing false sub domain because an attacker might manage to create a fabricated DNS entry which is a subdomain of the server, with the attacker’s server IP address. If victim clicks on attacker’s web address, since the attackers web address is subdomain of original web address, victim’s browser will submit to the attacker’s server all cookies including the username and password entered. | For markers' use only |
| How would you correct the problem? | by not displaying the username and password in the address bar and keep it hidden. Use POST instead of GET. Enforce a stronger policy when creating subdomains. Disable cookies. |  |

|  |  |  |
| --- | --- | --- |
| Describe the defect | In /var/www/html dosend.php, since the filetype uploaded will be jpg, it should not be executable therefore chmod should not be 700. If it is executable, attacker can upload executable files, which if the victim’s browser runs, attacker might gain control, resulting in cross-site scripting. Cookies will be sent to the attacker server. | For markers' use only |
| How would you correct the problem? | Website should filter out malicious code. |  |

|  |  |  |
| --- | --- | --- |
| Describe the defect | In /var/www/html login.php, if any user/attacker is able to gain access to name and password of a user who logs in to bank, they will be able to know their account details, such as sort code and account, as the login.php suggests. | For markers' use only |
| How would you correct the problem? | by not displaying account details after the user has logged in. Set a security question, to which if they answer correctly, account details can be displayed. |  |

|  |  |  |
| --- | --- | --- |
| Describe the defect | In /var/www/html/others/CS140 following a successful login get.php stores username and password in temp.txt file. Username and password are not hidden and are stored in plain text. This means that attackers can use this password to login to CS140 and access the CS140 webpage resources. | For marker’s use only |
| How would you correct the problem? | Store the password in a SHA 512. |  |

|  |  |  |
| --- | --- | --- |
| Describe the defect | In /var/www/html/others/CS140 CS140.php, following a successful login, (as code defined by login.php- setcookie(‘username’, $\_POST[‘username’],time()+36; setcookie(‘password, $\_POST[‘password],time()+36) displays username and password as input parameter of the request and stores it as a cookie. The form parameters are encoded in the URL, and the URL (upon clicking) appears in the address bar of the web browser. This is not a secure way of storing and sending the form data. It could easily be exploited. It also contains the IP address of the web server, which is sensitive information. Network eavesdropping can occur, in which cookies can be intercepted by attacker. Phishing false subdomain can occur because an attacker might manage to create a fabricated DNS entry which is a subdomain of the server, with the attacker’s server IP address. If victim clicks on attacker’s web address, since the attackers web address is subdomain of original web address, victim’s browser will submit to the attacker’s server all cookies including the username and password entered. | For marker’s use only |
| How would you correct the problem? | By not passing username and password as cookies, therefore by removing “?cookie=’+document.cookie’ and using POST method instead, which is secure and the contents of HTTP requests can be encrypted. |  |

**Please write the total number of security defects you found in the following box:**

|  |
| --- |
| **9** |