Report by - Shivank Garg(160658)

Securing the Platform

1. Securing the Platform for **sqli attack**-
    a. Use parameterized/prepared SQL
    b. Use the ORM framework
    c. use tools that automate the discovery of SQL injection flaws, and attempt to exploit SQL injection
    d. Monitor the platform for malformed input.
2. Securing the platform for “Game of Collision”
    a. Check the malformed input in PHP, so that it does not bypass the check by declaring the input to be an array at a later stage.
3. Securing the platform for “Headers speaks Loudly”-
    a. Getting admin rights by modifying a cookie is wrong in the server. This calls for(solution given in ref-3).
4. Securing the platform for **LFI and RFI attack**-
    a. Use of vulnerability scanner for LFI and RFI attacks, Like dorking or using automated tool which scans for potential vulnerability.
    b. Sanitizing the user input in php.
    c. Whitelisting the permitted files.
5. Securing the platforms for “**Traversal Attacks**”(Solution given in ref-5)-
    a. Prefer working without the user input.
    b. Whitelisting the acceptable input data.
    c. Use of chrooted jails to prevent unprivileged access.
    d. Use of Code access policies to restrict where the files can be obtained or saved to.

References-

1. https://www.esecurityplanet.com/threats/how-to-prevent-sql-injection-attacks.html
2. https://web.cse.iitk.ac.in/users/spramod/courses/cs628-2019/lectures/module-5.pdf
3. https://stackoverflow.com/questions/2848134/what-are-best-practices-for-securing-the-admin-section-of-a-website
4. https://security.stackexchange.com/questions/97183/preventing-lfi-with-user-input
5. https://www.owasp.org/index.php/File_System#Path_traversal