DESIDOC Journal of Library & Information Technology, Vol. 41, No. 5, Sept 2021, pp. 391-396, DOI : 10.14429/djlit.41.5.15975

OJS Security Analysis: Issues, Reasons, and Possible Solutions

Lakshmi Verma

DRDO-Defence Scientific Information and Documentation Centre (DESIDOC), Delhi – 110 054, India
E-mail: lakshmi24verma@gmail.com

ABSTRACT

Open Journal Systems (OJS), a modern era Publishing tool for authors, reviewers and editors have gained a lot of popularity in the recent times as this software is available free for use on web and publishes journal online. While this tool empowers its user to validate, support, control, track publications, etc, at the same time its wide user base has raised few concerns about data security. This article deals with security issues that may arise from the use of this web-based journal management and publishing software by the author and also suggests measures/precautions on how to minimise the possible risk related to data security based on author experience in certain situations. For this, the author has adopted a methodology that synchronises reviewed research papers with thoughts gained by reading various blogs and documentation and doing analysis of same. With this contribution from the author, the user is expected to benefit from the implementation of suggestive guidance/approach as prescribed in this article to overcome similar issues, which may be faced by some users. The author has endeavored to express the associated security issues, recommend solutions and security steps to be followed while using the OJS in certain situations.

Keywords: OJS; Security; Open journal software; Open journal systems; Defacement; Security issues

1. INTRODUCTION

With the ease of access to the Internet among all, the use of multiple applications/software which facilitates instant and easy communication and few being used for publication purposes has increased rapidly. While various software can be used for publication, at the same time it is important to ensure to choose the one which is easy to use, facilities in securing the desired output but with zero compromises on security aspects. The modern authors/editors consider one such software named OJS as a friendly resource which has also gained popularity as it is free to download and use and its convenient combat tools which support in the writing field. However, as it is a free open resource, there may be some possible security threats, which the user needs to understand before going forward with it.

Further, all authors, reviewers, and editors are availing the facility of this journal management software without knowing the complete benefits and few risks associated with such software. Thus software must be installed or customised in a way so that nobody can hack the information, deface any website or lead to any other possible risk.

Thus, a need is felt by the author of this article to throw light on the few security issues related to OJS. The author has worked on the analysis of the issues, their reasons, and related possible solutions to secure this Journal management software. The objective of the article is to provide security-related solutions to the user more sequentially and most easily for given points. The result gives the qualitative analysis of issues. This article focuses on the precautionary steps to be followed during the use of OJS in certain situations.

At present, a series of various latest versions[1] of OJS are available on the internet. All series have some standard safety features and it has been developed and released by Public Knowledge Project (PKP) since 2001.

PKP provides the most commonly used open-source journal publishing platform OJS and works both as a management and publishing system. It is freely available and can be downloaded and installed easily for the publication of the articles. With this journal's workflow can be tracked and keeps logs and work can be managed from author registration till the publication of the article in a systematic manner. The other key feature due to which it is being used so widely is that it reduces the time and energy related to the managerial and clerical work, improved the efficiency of the editorial process, made the journal policies more transparent.

The Yoris study suggests OJS 3.0 development and presents its effectiveness as a medium of learning resource for students, with ease of accessing the articles[7]. However, it may be noted that at the same time differences were observed in the effectiveness of the research on the development of OJS due to variation in results.

The Edger study suggests that open source tools have supported the revival of scholarly published peer-reviewed journals that are published worldwide and also available online[8].

John Willinsky mentioned the highly flexible editor–operated journal system, the open-source solution to online

Received : 28 May 2020, Revised : 23 May 2021
Accepted : 21 June 2021, Online published : 26 August 2021

manage and publish scholarly journals. It is also a cost, time, and energy effective method leading to low publishing cost and time as compared to other systems, for a managerial clerical task at the same time improves the qualitative aspect of the journal with more transparent policies[9].

Management with well-defined workflows for journals, open-access peer-reviewed journals can be easily supported and increasing productivity and visibility of Indian search. Besides, OJS is a multilingual system, which allows Journals for publishing in various languages[10].

R, Sreejith, designed, developed the independent website, and implemented the OJS for Rajagiri Journals; both Rajagiri Journal of Social Development (RJSD) and Rajagiri. The author discussed the challenges and their solutions during this implementation, they scanned the hard copies of all back volumes and successfully uploaded them on the OJS platform, and reduced the time required to manage a scholarly journal[18].

The author reforms the solutions available on blogs and websites and presents them forward in the article. However, some of them are available in various blogs on different websites as well.

Everything in the workflow process of publication can be done online, from submitting articles and reviews to the publishing of articles through OJS. On using any online publication system, any possible risk may impact the reputation of the publication. This brings the author to the point of the study of analyzing the types of possible security issues that can change the publication's website and defining the steps to the possible solution from the configuration to registration, submission, and publication process.

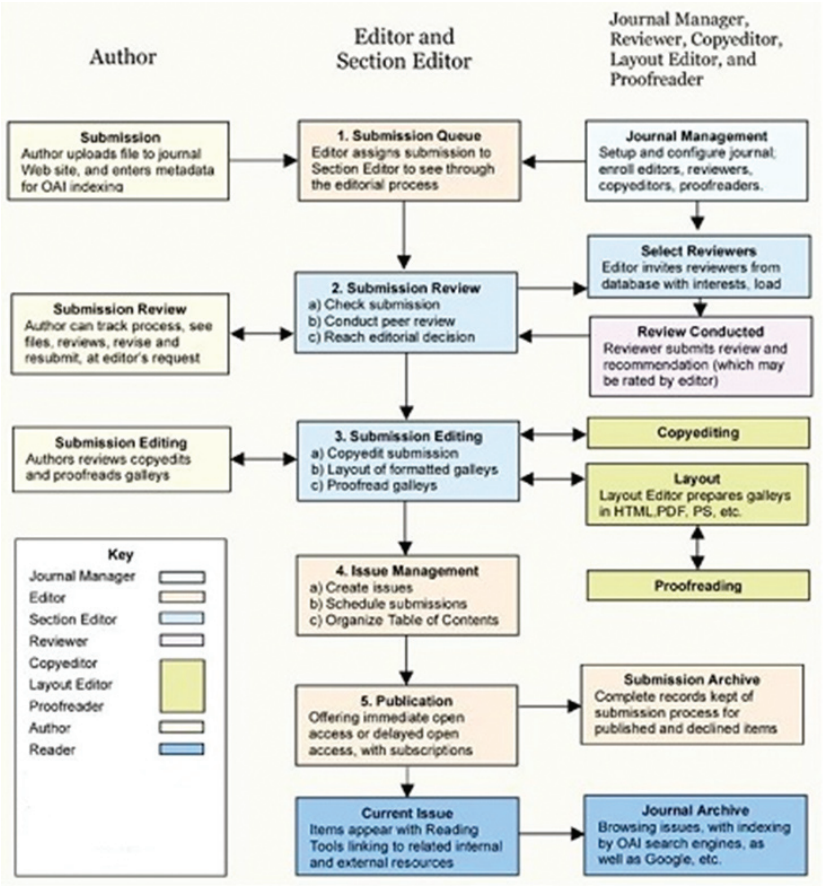

Figure 1. Editorial and publishing process in OJS.

2. METHODOLOGY

Various information such as issues, reasons, and challenges gathered from different ways and studied for the security of open journal systems. The author reviewed many research papers and browse different websites, forums[22] even though some were in different languages and collected the data for security issues and their solutions. Read various blogs and surveyed thoroughly various documentation for installation and safety issues to maintain the Open Journal Systems. Further studied online the different types of defacement issues through blogs and documentation and then find out their reasons and provided the possible solution based on the analysis.

3. WEBSITE DEFACEMENT OF OJS

There are various other Journal management and publishing software like Janeway, Ambra, available on the web, however, the author has focused on OJS due to its wide popularity and ease of use.

OJS software has already been used for publishing more than 10,000 journals worldwide and few examples of such Journals are, (International Journal of Engineering And Computer Science (IJECS); International Information and Engineering Technology Association (IIETA); Current Science Association, Indian Council of Agricultural Research (ICAR). Editorial and Publishing Process in OJS has been shown in Fig. 1.

Nobody knows until the Journal website have been hacked, here are few signs to be noticed that different landing page in OJS, content changed or funny character included blank pages or PHP errors. Defacement/attack on any online journal gives visitors a bad impression about the security and incapability, to handle the particular publishing system. This may leave visible marks in particular publications. There can be multiple reasons for defacement.

Based on the user's experience, some kinds of attacks/ defacements, their causes, and their solutions are mentioned as follows.

3.1 First Type of Defacement Issue

The attacker uploads the different index.php, Malicious .php file and also rename files /folders

3.1.1 Reasons

This is due to the perilous deployment of the Software. If the files directory is available through a web server, then the attacker can create a URL and access the PHP file directly to get it executed.

3.1.2 Possible Solutions

- Replace the OJS installation files to ensure that there are

no changes other than the modified index.php file. Most likely only to modify index.php
- Change database password
- After OJS goes online, it is worth checking that the hacker has not added any new users with manager privileges
- Files directory should not be created with sub-directory of the OJS installation so that they are not available for direct access
- You can use .htaccess rules to protect direct access of files directory as .htaccess rules control the access to files directory. With an insecure .htaccess rules configuration, a hacker can get unauthorised access to the publication[4]
- Only allow safe files extensions (i.e .pdf, odt, .doc, .docx) and prohibit the server-side executable files extensions (such as.php, .rb, .phtml, .asp,.py) to upload the files[2].

3.2 Second Type of Defacement Issue

Entry of unknown submission in the file directory. Downloaded file .phtml extension containing some text related to hackers.

3.2.1 Reasons

- Probably fake submissions by the author without completing it
- The problems are related to the .phtml file. which is uploaded during the submission.

3.2.2 Possible Solutions

- Install OJS in such a way that the files directory is not a subdirectory of the OJS installation, hence these cannot be accessed through a web server
- Use the .htaccess rules to protect direct access
- Read /Write permission should be given to the webserver only and the main file/ folder should not be readable by other users on the server[24]
- File/ folder permissions may be changed over time. A clever attacker can use the open folder to upload a PHP script that provides a back door to your site. It might not work at the moment but the attacker can wait
- Only allow safe file extensions (i.e .pdf, odt, .doc, .docx) and prohibit the server-side executable file extensions (such as.php, .rb, .phtml, .asp,.py) to upload the files.

3.3 Third Type of Defacement Issue

Intentionally misuse a submission feature to pretend as the hacking of publication.

3.3.1 Reasons

Like the other similar application systems, it allows users to register themselves as the author during the registration process, user can upload an image related to his profile. The application confirmed that the uploaded file is an image rather than what is in that image. So this is how defacer searches OJS Journal online, registered as an author (often with username "a"), and uploaded the "hacked by Mr. X " image to the "comments to the author section " on the submission page. Then they took the link (…./images/username/xyz.jpg) and finally they provide the direct link of the image to the various security agencies to pretend about the hacking of publication[3].

3.3.2 Possible Solutions

This is not defacement or hack though it has been reported as such by the perpetrators.
- First of all, delete the problematic file (image file) and the user account from the database
- Restrict the image file extension to prevent uploading of image profiles
- Access the history page of the user in OJS, so the manager can track down which users registered today or changed their profile
- Allow permission to get the approval of individual content changes (like changing profile pictures or biographical statements) by the manager
- Create a file directory at a different location from the directory of OJS installation
- Mediate the access through the website URL instead of the exact path
- The captcha should be enabled in config.inc.php (This will not stop all registrations, but it will slow down automated attempts significantly)
- Add the new restriction in the upload file method under modify lib/pkp/classes/file/FileManager.inc.php[3].

4. SECURITY STEPS TO BE TAKEN DURING CONFIGURATION/INSTALLATION OF OJS[15-16,23]

Due to the misconfiguration of OJS, the hacking risk may be high. Following steps can be taken to reduce it.
- Files directory should be at a non-web-accessible location of OJS installation[24]
- Ensure the installation and working of SSL certificate so that site always uses the HTTPS protocol to manage the user registration process, log in, and present the content to the readers[13]
- Set the long, random password, OJS database should be completely dedicated, credentials should be unique to access it
- Configuration of this database should be to perform automated backups regularly
- Before up-gradation or maintenance take a complete backup of database and files folder [6]
- Back up should be comprised of the OJS submission files directory, the OJS files, and the database.[9]
- Configure OJS to use SHA1 hashing rather than MD5[24].
- Most spam user registrations can be prevented by enabling captcha or Recaptcha in config.inc.php file and tested as they are working
- Web server environment should be regularly updated, in particular with any security patches. Restrict file permissions as much as possible. (Only allow safe extensions (i.e .pdf, odt, .doc, .docx) and prohibit the server-side executable extensions (such as.php, .rb, .phtml, .asp,.py) to upload the files)[5,12]

5. SECURITY STEPS TO BE TAKEN DURING UP-GRADATION OF OJS

- As per a study, up-gradation of OJS is very much required at regular intervals as it is open-source software, it improves its performances, functionality, productivity, compatibility. Since security is the main concern so before up-gradation of installation of OJS a backup of the database and other essential files are required[10]. Manual/ Automated backup can be performed for this. Upgradation of the OJS database can be done by command line or web interface[14,23-24]
- There should be secure file management as the author, reviewer, and editors work with submission files on a daily basis. Basic precautions to deal with it are given as follows:
  a) Ensured antivirus software installed and updated
  b) Operating system and all software keep updated
  c) Strong password management and if a submission paper has a suspicious title/ abstract/ e-mail address, then included files can be treated with an additional level of diligence
  d) To secure the site SSL/TLS encryption ensures the security and privacy advantages like- show warnings for insecure websites served under http. Admin and user login credentials can be scanned without encryption. User searching and accessing within the site are also not protecting without encryption
  e) A website security certificate enables a secure connection from the user's web browser to the server hosting site also ensures for maintenance of a certificate
  f) Spam can be managed through captcha/Recaptcha. Google's Recaptcha using public and private keys which is more secure
  g) Email validation steps must be completed for all new user account
  h) Cleaning of lots of users required by using the merge user options.

6. SECURITY STEPS TO BE TAKEN FOR A REGISTRATION PROCESS IN OJS

- User needs to duly fill the given registration form with all required fields and he has to agree with the policy documents related to his collection and stored data as well
- Password length should be considered to get it strong including lowercase and uppercase alphabetic characters, numbers, and symbols. Then the user receives an email with a link to validate his email account
- To reset the password, on clicking the link, the user receives an email with a username and new password in plain text. If the manager does register any user, then the user receives an email with his username and his new password in plain text. It is not secured to send a password in plain text, it would be better to send the user an email with a non-replayable link that allows the user to complete a new password online with a link to validate his email account as in self-registration.

7. HTACCESS OR SIMILAR MECHANISM TO PROTECT DIRECT ACCESS OF DIRECTORY

Files directory must not be placed in a web-accessible location if it is placed at a web-accessible location then it needs to be protected from direct access, such as via .htaccess file rules. A .htaccess file is a configuration file and it allows the configuration setting for a particular directory. It can include one or more configuration settings to apply only for the directory in which the .htaccess file has been placed. So while the web servers have a main configuration settings file, the .htaccess file can be used to override their main configuration settings. Each directory can have its .htaccess file.

The .htaccess file controls the access of this root directory and eventually gain access to all the subdirectories[19-20].

7.1 Prevent Direct Access of all the Files and Folders

Create a .htaccess file in the OJS installation directory and write "deny from all" in .htaccess file[11]. Following messages (Forbidden/ You don't have permission to access) will be displayed on trying to access any directory.

7.2 Create Password Protect Directory through .htaccess

Create one .htaccess file in a directory, which you want to protect with a password. Write the codes in the .htaccess file to create a password to protect the directory/ deny access to certain file types/ prevent any direct directory browsing, as documentation available on the internet.

8. PLUGINS

8.1 OJS File Upload Validation Plugin

OJS File Upload Validation Plugin has been developed specifically to address OJS security vulnerabilities. This Plugin allows the Administrator to choose the file types for uploading during the submission process. It prevents the malware, i.e PHTML shellcode files to upload by which hackers gain access to an OJS web server. Usually hackers insert the * .php or * .html to do deface activities. These must not be uploaded to OJS. The editor can choose the file extension that is allowed in the journal.

8.2 OJS Registration Notification Plugin

Whenever a user register for a journal an email notification is sent to a predetermined address through this plugin. This e-mail includes the journal name, registrant, and user role. An administrator can verify the new user and delete the fake user's registration long before they can do any damage.

9. GOOGLE RE-CAPTCHA

CAPTCHA is a challenge-response test technique on computers that serves to prevent filling in the form with a computer robot, with the CAPTCHA, the system will be safer. google: reCAPTCHA, can be accessed via https://www.google.com/recaptcha/intro/v3.html

After login Google account, select reCAPTCHA and register the site in the given form. Choose the Type of

reCAPTCHA. The domain is the website address that is registered. After checking the agr ement, register the site key, the secret key will appear to be added to the OJS. Then edit the config.inc.php file[17,21] and register as an author for a journal.

10. ANALYSIS

For types of defacement issues, security steps shall be taken during configuration/ installation, up-gradation, and registration process. Modify the file permissions of OJS, location of public and root folder, Google Recaptcha to prevent direct access to all the files and folders through .htaccess or similar mechanism. For validation and notification plug-in, access through a different path instead of direct URL, which have been analysed and provide the possible solutions.

11. DISCUSSION AND RESULT

The content of mentioned references has been studied and it was found that different issues were in the various resources and the same issues have been represented in different ways so the user needs to give more attention and efforts for the solution of the similar issues. The author provides the solution to described issues and copes up with challenges like were in different resources, in multiple ways, and in different languages. The Author identified and after analysis, displayed them in a simpler way. The author used online tools available for translation to provide some of those solutions. There are so many issues and vulnerabilities that have been left that are not presented under this article those have to be identified further. The author studies and understands the multiple parameters of the issues from multiple sources and described them in the easiest way. Initially found the issues and their solutions secondly provided security procedure that user can follow during installation, during up-gradation of OJS, during the registration process, on shifting of the public folder, during preventing the direct access of files directory, on insertion of plugins, while the accession of URL, on avoiding the computer robot during filling up the form.

12. CONCLUSION

Security of the OJS cannot be done exclusively as it involves different procedures/ steps to be adopted at different stages. Thus, it must begin from the installation of the OJS software to the configuration of the OJS software. It must further be managed by admin permissions for the user's registration. This paper provides some possible solutions to the security-related issues of OJS. In this article, these issues and their remedies have summed up understandably in described situations.

REFERENCES

1. PKP, Open Journal system, Download, https://pkp.sfu.ca/ojs/ojs_download/ (Accessed on 16/11/2019).
2. Securing your system. https://docs.pkp.sfu.ca/admin-guide/en/securing-your-system. (Accessed on 07/12/2019).
3. Regarding recent OJS "Defacement" attacks. https://pkp.sfu.ca/2017/04/12/regarding-recent-ojs-defacement-attacks/ (Accessed on 11/12/2019).
4. Fix identified security issues to avoid breaches. www.websitedefender.com (Accessed on 20/12/2019).
5. WordPress security. www.siteground.co.uk (Accessed on 3/11/2019)
6. FAQs. https://github.com/pkp/ojs/blob/master/docs/FAQ. (Accessed on 21/01/2020).
7. Yoris, Adi Maretta; Sumaryanto, Totok; Utanto, Yuli; Kustiono, Kustiono & Deliana, Sri Maryati. Development of Open Journal System 3.0., 2018, **205**(00007). doi: 10.1051/matecconf/201820500007.
8. Edgar, B.D. & Willinsky, J. A survey of the scholarly journals using Open Journal Systems. 2010, **1**(2). doi: 10.7146/ojssb.v1i1.2707.
9. Willinsky, J. Open Journal Systems: An example of open source software for journal management and publishing. *Libr. Hi. Tech.*, 2005, **23**(4), 504–519. doi: 10.1108/07378830510636300.
10. Vaidya, Nagaraj; Obaiah, B. & Thomas, Abraham. Open access journal publishing in India: A study with OJS software. 2009, 479-484.
11. OJS 3 Security–Files Directory. https://ilmubersama.com/2018/11/17/ojs-security-files-directory/. (Accessed on 13/12/2019).
12. File permissions tutorial. https://www.siteground.com/tutorials/cpanel/file-permissions/. (Accessed on 06/02/2020).
13. Security. https://libyanspider.com/ (Accessed on 03/01/2020).
14. Open Journal Systems, The Public Knowledge Project. https://github.com/pkp/ojs/blob/master/docs/README.md. (Accessed on 08/02/2021).
15. Recommended configuration. http://www.publishmed.com/docs/README. (Accessed on 02/02/2020).
16. OJS hosting requirements–OJS manual installation. https://www.tmdhosting.com/kb/question/ojs-hosting-requirements-ojs-manual-installation/. (Accessed on 04/11/2019).
17. Salt (cryptography). https://en.wikipedia.org/wiki/Salt_(cryptography). (Accessed on 08/04/2020).
18. Sreejith, R.; Vijayan, Vijesh; & Francis, A.J. "Design and implementation of Open Journal System (OJS) for Rajagiri Journals: A Review", 2019.
19. Security precautions when using .htaccess Files, https://www.acunetix.com/blog/articles/what-is-an-htaccess-file/. (Accessed on 09/12/2019).
20. Using .htaccess. https://docstore.mik.ua/orelly/linux/apache/ch05_11.htm. (Accessed on 18/12/2019).
21. What is a salt and how does it make password hashing more secure? https://www.skyhighnetworks.com/cloud-security-blog/what-is-a-salt-and-how-does-it-make-password-hashing-more-secure/. (Accessed on 21/03/2020).
22. PKP community forum. https://forum.pkp.sfu.ca/. (Accessed on 06/11/2019).
23. Upgrading an OJS installation. https://github.com/pkp/ojs/blob/master/docs/UPGRADE.md.

(Accessed on 10/02/2021)
24. Deploying PKP software securely. https://docs.pkp.sfu.ca/admin-guide/en/securing-your-system#deploying-pkp-software-securely. (Accessed on 05/02/2021).

CONTRIBUTOR

Ms Lakshmi Verma working as a Technical Officer in DRDO-DESIDOC, Delhi. She has received her MSc (Computer science) from MDU, Rohtak. Her area of interest include electronic resource management, digital archiving, content management systems.