International Journal of Information Technology and Knowledge Management
January-June 2011, Volume 4, No. 1, pp. 233-238

INVESTIGATION ON SECURITY IN LMS MOODLE

Sheo Kumar[1] & Kamlesh Dutta[2]

E-learning provides the opportunity to student to interact electronically with each other as well as with their teachers. This interaction can be via e-mail or on discussion board or in chat rooms. Though recognizing that the world at large will persist to use language and terminology in different ways, so the term of virtual learning environments (VLE) is used to refer to the on-line interactions of a variety of kinds that take place between students and teachers. There are many software systems available that provide VLE systems. This software is in both forms, commercial and open source software (OSS). Moodle is the one of them systems that has been increasingly gaining worldwide popularity in e-learning system.

LMS Moodle has much vulnerability like authentication, availability, confidentiality and integrity attacks. So, it is necessary to develop a mechanism that defends these security flaws of LMS Moodle. We present most common security flaws and suggest optimal security settings of Moodle (Modular Object-Oriented Dynamic Learning Environment) LMS and the server itself. Specifically, we will focus on authentication attack from above mentioned flaws. We further classify design and session attack. Design attack on moodle can be broadly categorized as password prediction and user name prediction. Session attack on moodle is session hijacking. Moodle is an open source software e-learning platform that provides educators tools to create a course web site. Through the last few years, LMS Moodle imposed itself as the best solution, and is becoming one of the most common used systems.

The open source learning management, LMS Moodle has been adopted by many people and organizations around the world because it offers a tightly integrated set of tools said to be designed from a social constructive perspective. Moodle has been developed under the general public license and many of its components were developed without a specific design documentation including its security services. Object oriented model of Moodle using an analysis of its security services as well as solutions to its security vulnerabilities.

Keywords: Moodle, Virtual Learning Environment, Open Source Software, Web Attack, Security

1. Introduction

Together with the fast growing popularity of the Internet in current years, there is a growing demand for methodologies and technologies for e-learning. E-learning is an interactive learning in which the learning content is accessible on-line and offers automatic feedback to the student's learning behavior [1]. Therefore, there has been a growing demand for VLE methodologies and technologies. VLE (virtual learning environment) is defined as interactive learning in which the learning content is offered on-line and offers automatic feedback to the student's learning behavior. While knowing that the world at huge will carry on to use terminology in different and frequently unclear ways, the word of VLE is used here to refer to on-line communications of different kinds including on-line learning that takes place between learners and instructors [2, 3]. At present, there are previously more than 250 sources of commercial e-learning and more than 45 of them are Open Source Software (OSS) offerings as free of charge VLE systems. The better-known OSS are Moodle, Ilias, eduplone, Claroline, SAKAI, WebCT and Bscw, and they have wide developer societies who present strong points of view for considering OSS as a straightforward and potentially practicable competitor to commercial products. Modular Object-Oriented Dynamic Learning Environment (Moodle) is an OSS project that has appeared to meet the rising interest in OSS. Moodle is a web-based Learning Content Management System (LCMS), i.e. a Course Management System (CMS) and VLE planned around pedagogical principles, namely a social constructivist viewpoint using the collaborative possibilities of the Internet. It permits teachers to present and allocate documents, assignments, quizzes graded, etc. with students in an easy-to-learn way, and to generate quality on-line courses. Moodle is a free OSS, which means user are free to download, use, adapt and even to distribute it under the terms of GNU[4,5]. A significant resource for higher education, particularly universities, is VLE, which has been attractive students' progress with high quality learning around the world.

The LMS called Moodle [6], is open source software, and can be configured to run on most operating systems (Macintosh OS, Windows XP and Linux). Moodle was developed from a social constructivist viewpoint by Martin Dougiamas at Curtin University in Western Australia. Moodle has some features like including the capability to embed resources and communication and activities centered on a topic of study, which is not available elsewhere. The trainer may also specify a variety of modes for operation

[1]Assistant Prof., Dept. of CSE, RKGITW Ghaziabad (UP), India

[2]Associate Prof., Dept. of CSE, NIT Hamirpur (HP), India

Email: sheokmr@gmail.com[1], kdnith@gmail.com[2]

(from weekly formats, topic-based to social formats). The acceptance and implementation [7] of Moodle has been extremely successful. The uptake of the software has been so successful that the first user conferences (called MoodleMoots) took place in July 2004. MoodleMoots have been held in the United Kingdom, Germany, Ireland and the USA. Moodle is also beginning to challenge the dominance of the pay-as-you-go model of many of the commercial LMSs available.

2. Reason for Choosing Moodle

Moodle has proved its importance due to a wider acceptance in the community and number of institutions. The software provides support for large number of courses in different languages [8, 9]. Moodle allows users the ability to post news items, assignments, electronic journals and resources, and to collect assignments etc. The community has grown around the project; both developers and users participate in Moodle's active discussion forums, sharing tips, posting code snippets, helping new users, sharing resources and debating new ideas which add strength to the Moodle [10, 11, 12].The most important reasons for choosing Moodle are listed below:

1. It is an OSS, which means users are free to download it, use it, modify it and even distribute it under the terms of the GNU license [2, 4, and 13].
2. It is a CMS & VLE that lets teachers provide and share documents, graded assignments, discussion forums, etc. with their students in an easy-to-learn fashion, and in high quality on-line courses [4, 14].
3. Moodle can be used on almost all servers that can use PHP. Users can download and use it on any computer and can easily upgrade it from one version to the next [11, 15].
4. The key to Moodle is that is developed with both pedagogy and technology in mind. One of the main advantages of Moodle over other systems is its strong grounding in social constructionist pedagogy and good educational tools [16].
5. The Moodle software is used all over the world by independent teachers, schools, universities and companies. The credibility of Moodle is very high. Currently, there are 3324 web sites from175 countries that have registered with it, and it has 75 languages [11, 4].
6. Moodle runs without modification on any system that supports PHP such as Unix, Linux and Windows. It uses MySQL, PostgreSQL and Oracle databases, and others are also supported [13].
7. It has many features useful to potential students such as easy installation, customization of options and settings, good support/help, and good educational tools. Moreover, it has excellent documentation, and strong support for security and administration [16].

3. Limitations of Moodle

Moodle is ease of use, flexibility, low cost, and helps bring VLE technology within the reach of those with limited technical or financial resources [9]. On the other hand, Moodle has some limitations as follows:

1. Moodle is only for IT experts. It is complex for normal users to use and more than 66% of them are teachers, researchers and administrators [11]. It is difficult for beginner technicians to install and use Moodle [9], because there are many technical word lists in installation instructions.
2. Moodle will work, but not by itself. If there is not a course administrator that can work with both teachers and technicians in creating on-line materials, then Moodle will remain an empty shell, like a good aircraft but with no pilot. Lack of simple-to-obtain support [11]. Forums carry a great deal of information, but nearly all forums are in the English language.
3. It does not support the SSL implementation all over the site.
4. It stores the user data into cache which can be later used by the attacker to launch the attack for next session.

Brute force attack is possible on Moodle as the attacker may try different keys for several numbers of times.

4. Security Vulnerability in Moodle

In this paper we present description of the most critical security flaws as discussed in literature. They are classified into four groups: authentication, availability, confidentiality and integrity attacks. Learning management systems [17, 18] are client/server web applications that, among rest, handle user requests coming from clients such as web browsers. To handle the user requests, they frequently require accessing security-critical resources (databases and files) at the server end.

Table1
Attack Methods and Security

i)	Authentication attacks
1.	Broken authentication and session management
2.	Insecure communication
ii)	Availability attacks
1.	Denial of service

iii) Confidentiality attacks

1. Insecure cryptographic storage
2. Insecure direct object reference
3. Information leakage and improper error handling

iv) Integrity attacks

1. Buffer overflow
2. Cross Site Request Forgery
3. Cross Site Scripting
4. Failure to restrict URL access
5. Injection flaws
6. Malicious file execution

Table 1 shows a summary of confidential attack methods and vulnerabilities independent of the correct LMS implementation as presented in [17]. Model used to group attack methods and security vulnerabilities is widely accepted AICA (Availability, Integrity, Confidentiality and Authentication) threat modeling approach.

4.1. Authentication Attacks

Authentication and session management includes all aspects of handling user authentication and managing active sessions. Authentication is a dangerous feature of this process, but even hard authentication mechanisms can be damaged by flawed credential management functions, including password change, forgot my password; remember my password, account update, and other related functions. Vulnerability which appears during transmits of responsive information (session tokens) without proper encryption. Attacker can misuse this flaw to impersonate user and access unprotected conversations.

4.2. Availability Attacks

The main objective of availability attacks is to make e-learning services and data unavailable to authorized end users. Most popular variety of availability attack is denial of service (DoS) attack. A denial-of-service attack (DoS attack) [19] is challenge to make a computer resource unavailable to its intended users. Although the means to carry out, motives for, and targets of a DoS attack may vary, it generally consists of the concerted efforts of a person or people to prevent an Internet site or services from functioning efficiently or at all, temporarily or for an indefinite period. Perpetrators of DoS attacks are not limited to services hosted on high-profile web servers such as banks, credit cards payment gateways; it is also used in position to CPU resource management. There are generally two types of DoS attack, first are logic attacks and second are flooding attacks. Logic attacks exploit existing LMS flaws to crash remote server or significantly decrease its performance. Flooding attacks overloads LMS with a high number of requests to disable legitimate users from accessing e-learning resources.

4.3. Confidentiality Attacks

Confidentiality attacks are submissive kind of attacks which allows prohibited access to confidential resources and data. The main purpose of attacker is not data alteration but data access and distribution. The most often confidentiality flaws are: Insecure cryptographic storage, Insecure direct object reference and Information leakage and improper error handling.

Insecure cryptographic storage flaws, [17,18] which is based on a fact that sensitive information does not have appropriate encryption. Insecure direct object reference usually occurs when LMS uses object references directly in web interfaces without authorization checks being implemented. Mentioned object references can be files, database records and primary keys and are contained either by URL or form parameters. Information leakage and improper error handling refers to unintentional disclosure of sensitive data and unneeded information through error messages. LMS can leak sensitive information about its logic, configuration and other internal details (e.g. SQL syntax, source code, etc.). LMS systems not often use cryptographic functions properly to protect data and qualifications or use weak encryption algorithms. In both situations, valuable data is relatively easy to access by attacker who can conduct identity theft and similar crimes.

4.4. Integrity Attacks

This group includes attacks which attempt to create new data or modify and even delete existing e-learning data. Integrity attacks are: Buffer overflow attack, Cross Site Request Forgery (XSRF/CSRF), Cross Site Scripting (XSS), Injection flaws, Malicious file execution and Failure to restrict URL access.

Malicious file execution attack [17, 20] which is based on a fact that LMS fails to control or disallow execution of uploaded files. Malicious code is usually uploaded via upload feature (e.g. homework or image). This kind of vulnerability can be found in many web applications, especially in those which are PHP based. Injection flaw may happen when data provided by user (e.g. in form fields) is sent to content checking routines as part of a command or query. In such attacks, interpreter fail to detect or respond to character sequences that may be interpreted wrongly, which then results in execution of malicious code by LMS. Finally, attacker could be able to create, update, read or delete all data available to LMS. Cross Site Scripting (XSS) [17, 21] refers to hacking technique which allows an attacker to supply vulnerable dynamic web page with malicious script and execute script in victim's browser in order to gather data from a user. Cross Site Request Forgery (XSRF/CSRF) is client side Attack which exploits faith that a LMS has for the user. When a user is logged into LMS, attacker can deception his browser into making a request to one of

LMS task URLs which will cause a change on the server. Buffer overflow attack [17, 22] occurs when a LMS module (e.g. libraries, drivers, server components) tries to store data into an available buffer without validating its size by inserting larger values than expected. Failure to restrict URL access, some LMS resources is limited to a small subset of advantaged users (e.g. administrators). This weakness allows an attacker to retrieve URLs by guessing the address and perform illegal operations on defenseless LMS data.

5. Security Attack to Moodle

With increasing demands of internet services results in the emergence of multiple service providers which provides access point to internet users. As there are tremendous increments in internet users recently, security issues have become the major concern. We have determined following security attacks on Moodle such as session attack; design attack and user log out, session not closed. Session attack which is effective against Moodle is session hijacking. As per the concern of design attacks, Moodle is vulnerable to password prediction and user name prediction. Another, security vulnerability is that when the user logout still the session is not closed. When the user clicks on the back button then he reaches the page which was logged out earlier.

5.1. Design Attacks

Moodle (modular object oriented dynamic learning environment) is defenseless to password prediction and username prediction.

5.1.1. Password Prediction

Brute force attack [29] can be performed by using the design flaw of Moodle server. To do this attack, the user sends the several requests to the Moodle server with the blank cookie field so that the login failure count is reorganize to zero when the cookie field is blank in the request.

5.1.2. User Name Prediction

This may be done by brute force method. Brute force attacks may be performed as like in password prediction. However, instead of sending several requests with different password, a number of usernames are sent with an arbitrary password. The response from Moodle will take longer with a valid username than with an invalid one and this was used to differentiate between them in the attacks realized.

5.2. Session Hijacking

The Session Hijacking attacks [24] consist of the utilization of the web session control mechanism, which is usually managed for a session token. Because http communication uses several different TCP connections, the web servers require a method to be familiar with each user's connections. The most helpful method depends on a token that the Web Server sends to the client browser after a successful client authentication. A session token is in general composed of a string of variable width and it could be used in different ways, like in the URL, in the header of the http requisition as a cookie, in other parts of the header of the http request, or yet in the body of the http requisition. The Session Hijacking attacks co-operate the session token by stealing or guessing a valid session token to gain unauthorized access to the Web Server. A hacker can also be in line between client and server using a sniffing program to watch the conversation. This is known as a man in the middle attack.

Session hijacking is a part of the eavesdropping attacks. Where an attackers pay attention the communication between client and server. They are trying to find inside the pay load, in this case the HTTP requests. The information that can be used to impersonate the user and attractive organize of his or her session. Moodle handles its session through two values to recognize an active session: MoodleSession and MoodleSessionTest.

MoodleSession and MoodleSessionTest principles are stored in the cookie hat is thrown on each HTTP demand within the header of the message. Acquiring a full HTTP request data with the cookie incorporated is easy because Moodle only uses SSL tunnels on the login service and a few administrative services. Because of that, most HTTP demand is done on plaintext that can be catch and easily decoded. After getting the cookie, the attacker can utilize this data on its own HTTP request, taking control of the target user session.

6. Proposed Solution to Moodle Security Attacks

In this paper we propose solution for the security attacks that are identified in previous section. For example when a user logout from a session then he/she is redirected to account if someone click the back button. By doing this, user goes to the previous active page, and can access the account, which is a big flaw in Moodle. To remove this flaw session can be used, when the session is used and the user login to the system his session becomes active and will remain active for the time he does not press the logout button. When the logout button is pressed his session expires and the user becomes inactive.

6.1. Login with CAPTCHA

When a user wants to login to Moodle server the brute force attack occurs. To remove this flaws CAPTCHA can be used which generates some random values that allows the user to enter these random values during his or her login. Authentication phase becomes stronger against the brute force attack by using this technique. The login page is combined with the official captcha completion known as recaptcha. Captcha implementation designs the random code generator algorithm. Design php code using graphics library to generate Captcha image for above generated code. Upon pressing the login button, it matches the security code entered by user and images.

6.2. SSL (Secure Sockets Layer) to Avoid Session Hijacking

SSL (Secure Sockets Layer) is the solution to avoid session hijacking problem. SSL is the standard security technology for establishing an encrypted link between a web server and a browser. Moodle already has a choice for using SSL over certain critical actions. However such method cannot avoid session hijacking and user name prediction. In order to avoid such attacks, the entire site must create SSL connections with its clients.SSL (Secure Sockets Layer)link ensures that all data passed between the web server and browsers remain private and integral.SSL (Secure Sockets Layer)can be done by adding a PHP scripts that changes the content of the object that holds the environment configuration named CFG. In CFG Themewww, Wwwroot, Login https, Https theme these are the following four variables that are SSL related. SSL is an industry standard and is used by millions of websites in the protection of their online transactions with their customers. To be capable to generate an SSL connection, a web server needs an SSL Certificate. When you decide to activate SSL on your web server you will be encouraged to complete a number of questions about the identity of your website and your company. Your web server then generates two cryptographic keys a Private Key and a Public Key.

7. CONCLUSION AND FUTURE WORK

In this paper, Moodle (Modular Object-Oriented Dynamic Learning Environment), an open source software e-learning platform is studied and security related issues of Moodle are discussed. Moodle provides educators tools to create a course web site. There are many security issues like as authentication, availability, confidentiality and integrity attacks is investigated under this work and specially an authentication attack from the above issues is carried out throughout the paper. Further, in authentication the session attack and design attack problems are rectified by using SSL (secure socket layer) and login with Captcha implementation respectively.

Whenever a user wants to login to Moodle server, the brute force attack occurs. Captcha implementation is used to avoid design attack problem, and Captcha technique which generates some random values that allows the user to enter these random values during his or her login. Authentication phase becomes stronger against the brute force attack by using this technique.

SSL is preferred to avoid session hijacking Problem. SSL is the standard security technology for establishing an encrypted link between a web server and a browser and SSL can be done by adding a PHP scripts that changes the content of the object that holds the environment configuration. Login https is used for secure data transmission. For implementation of such technique SSL certification is required.

To create more secure and reliable learning environment, it is essential to remove all the security flaws of the Moodle. In this paper, we focus on the authentication attack. This work can be extended by addressing other security issues of the Moodle.

REFERENCES

[1] S. Graham, "Building Web Services with Java: Making Sense of XML, SOAP, WSDL and UDDI", 1st ed., Pearson Education, pp. 450, 2001.

[2] G. Tortora, "A Multilevel Learning Management System, in: Proceedings of the 14th International Conference on Software and Knowledge Engineering", ACM: Ischia, Italy, pp-403-411, 2002.

[3] D. Martin, and T. Peter, "Interpretive Analysis of an Internet-based Course Constructed using a New Courseware Tool called Moodle", Quality Conversations in: Proceedings of the 2002 Annual International Conference of the Higher Education, Perth, Australia, pp. 560-473, 2002.

[4] http://www.Moodle.org.

[5] G. Sabine, and L. Beate, "An Evaluation of Open Source e-learning Platforms Stressing Adaptation Issues", in: Proceedings of Fifth IEEE International Conference onLearning Technologies, IEEE, Ischia, Italy, 2005.

[6] A. Bucher, "Moodle Administration: An Administration Guide to Configuring, Security, Customizing and Extending Moodle", Packt, 1-357, September 2008.

[7] K. Brandl, "Are you Ready to Moodle, in Language Learning/Technology", Washington, 9, No. 2, pp. 16-23, 2005.

[8] J. Cole, and H. Foster, "Using Moodle: Teaching with the Popular Open Source Course Management System", 2nd ed. O'Reilly, 2007.

[9] B. Williams and M. Dougiamas, "Moodle for Teachers, Trainers and Administrators of Remote-learner.net", Retrieved from http://Moodle.org.

[10] M. Zenha-Rela and R. Carvalho, "Work in Progress: Self Evaluation through Monitored Peer Review using the Moodle Platform", in Frontiers in Education Conference, 36th Annual., San Diego, CA: IEEE, pp. 230-241, 2006.

[11] A. Chavan and S. Pavri, "Open Source Learning Management in Moodle", in: Linux Journal, 1, No. 2, pp. 78-97, 2004.

[12] J. Itmazi, "Flexible Learning Management System to Support Learning in the Traditional and Open Universities", PhD Thesis, Granada University, Spain, 2005.

[13] S. Shearer, "Open Source Software in Education", The Compton School: London, 2003.

[14] M. Berry, "An Investigation of the Effectiveness of Moodle in Primary Education", in: Proceeding of IADIS Internet Conference, pp- 51-58, 2008.

[15] M. Dougiamas, "Moodle: Virtual Learning Environment for the Rest of Us", in: TESL-EJ (Teaching English as a Second or Foreign Language), 8, No.2, pp. 1-8, 2004.

[16] C. Su, "An Open Source Platform for Educators", in: Proceedings of the Fifth IEEE Advanced Learning Technologies, IEEE Computer Society, pp. 311-320, 2005.

[17] Z., Stapić, T. Orehovački, M. Đanić, "Determination of Optimal Security Settings for LMS Moodle", in: Proceedings of 31st MIPRO International Convention on Information Systems Security, 5, pp. 84.-89., Opatija, 2008.

[18] http://wiki.developers.facebook.com/index.php/Platform_Security.

[19] http://en.wikipedia.org/wiki/Denial-of-service_attack

[20] http://www.cgisecurity.com/csrf-faq.html

[21] http://en.wikipedia.org/wiki/Cross-site_scripting

[22] http://en.wikipedia.org/wiki/Buffer_overflow

[23] http://tldp.org/HOWTO/SSL-Certificates-HOWTO/x64.html

[24] http://www.owasp.org/index.php/Network_Eavesdropping