International Journal of Advanced Information Science and Technology (IJAIST) *ISSN: 2319:2682*
Vol.5,No.3, March 2016 **DOI:10.15693/ijaist.2016.v5.i3.53-61**

Secure Software Deployment: *Investigating the Security Vulnerabilities of MOODLE LMS in Public Higher Learning Institutions in Tanzania*

Mr. Said Ally
Department of ICT, Faculty of Science & Technology
The Open University of Tanzania, Dar Es Salaam, Tanzania

***Abstract*—Despite the wide spread adoption of MOODLE in Tanzania, there is still lack of software security implementation. This study establishes more evidences of security loopholes the software might undergo as a result of inefficient customization and management.**

The paper exposes security vulnerabilities of MOODLE platform by screening the internal source codes, software upgrading process and the adoption procedures using case study methodology. Four institutions were visited with 87 respondents categorized as Top Management, MOODLE Developers, MOODLE Administrators and MOODLE End-Users.

This paper discloses the extent of risk that can likely hinder adopting organization from attaining full intended benefits of using MOODLE software as LMS platform of choice.

The study reveals that there are lots of security weaknesses in locally customized MOODLE systems as a result of uncoordinated operations and *ad hoc* performance of key MOODLE stakeholders from adoption throughout software management. The most common vulnerabilities are Cross Site Scripting (XSS), Security Bypass, SQL Injection, CSRF and PHP Code Injection. Terribly, the password salt is not utilized and therefore the MD5 hash algorithm can easily be broken in MOODLE.

Index terms - Open Source Software, Security, MOODLE, e-learning, Tanzania, Web Vulnerability

I. INTRODUCTION

A. MOODLE as an Open Source LMS

The MOODLE is an Open Source Learning Management System (OS-LMS) which is freely available from the internet repository and is offered under the General Public License (GPL-GNU license) [1]. It requires a LAMP platform running Linux, Apache, MySQL and PHP but also can be configured to run on other operating systems including Macintosh OS and MS Windows [2], and MySQL, Oracle and PostgreSQL databases [3]. MOODLE can be redesigned to suit specific requirements of the adopting organization. Being free open source software, MOODLE can be downloaded, installed, used and distributed under the terms of GNU [4, 5].

B. The Usage Statistics of MOODLE

As of June 2013, MOODLE (*Modular Object-Oriented Dynamic Learning Environment*) had a user base of 83,008 registered and verified sites, serving 70,696,570 users in 7.5+ million courses with 1.2+ million teachers [6]. As of 24th March 2016, the MOODLE stats page had active registered sites of 70,136 serving 83,975,447 users across the globe (222 countries) with 9,435,080 developed courses and total enrolments of 243,171,965. MOODLE has high credibility and has been built to support large number of courses in different languages (over 75) [7, 8].

In Tanzania, several public HLIs have already invested in MOODLE as their e-learning platform. Since majority of HLIs depend on government as their main source of funds, the usage of OSS products like MOODLE has been a viable solution to them. With OSS tools, the developing countries can manage to leap frog and address the existing digital divide [9].

The web systems which are of open source nature like MOODLE are designed with open source codes, and so are accessible to both good guys and bad guys for software support and upgrading. With this fact, MOODLE turns to be of high potential risks from malicious attacks that can possibly introduced due to systems weaknesses and/or through deliberate and accidental actions by MOODLE users. Due to small budget allocation, most of the expenditures in MOODLE investment has been in training and content development. Very little or nothing is allocated for MOODLE customization and enhancement. The decision makers contend that MOODLE is downloaded with all system functionalities ready for use. So, based on this valid but insecure assumption, one can easily argue that if the HLI deploy and install MOODLE, it depends only on the preconfigured MOODLE security settings which are an indication that the software is exposed to attack and cannot be trusted unless the secure software adoption processes are observed and maintained.

Being a web based system of open source nature with open source codes, studying MOODLE security strength and its vulnerabilities provides special interest among researchers. As argued by [10], the security of these systems is a growing

concern for adopters. They may have financial implications or damage of public image of the adopters.

Therefore this paper presents investigations of the practices by HLIs in Tanzania towards MOODLE deployment. The interest is on how MOODLE is adopted, the extent of MOODLE customization and its relevancy in maintaining software security.

The remainder of the paper is organized as follows: section 2 presents theoretical knowledge of related work with focus on MOODLE security. Section 3 elaborate research methodology while section 4 discusses results and discussion. In section 5, concluding remark for further work is formulated.

II. RELATED WORK

Though application and use of Information and Communication Technologies (ICT) is growing very fast in the world; in developing countries like Tanzania, the use and development of ICT capabilities is still limited and faces a wide range of constraints and challenges [11]. One example of the areas which demonstrates such ICT immobility in Tanzania is the security of the web based Open Source Information Systems (OSIS). The HLIs regardless of their delivery mode (conventional or Open and Distance Learning (ODL)) have begun to consider the use of MOODLE as a key part of the e-learning strategy. In developing countries in particular, with the resource constraints they face, HLIs view MOODLE as a means of reducing the cost of e-learning investment. The imperative to adopt open source products particularly in the public HLIs is also motivated by a desire for independence, a drive for security and autonomy and a means to address intellectual property rights enforcement [12]. In Tanzania, a study conducted by [13], shows that there has been a growing interest towards application and use of open source systems which is an exclusive technology that has become an essential part of providing the fastest and most efficient organizational solutions [14].

Of interest, despite the mass adoption of MOODLE software, most of the MOODLE research studies are focused only on the experience in usage and extent of investment. There are limited number of researches that have been reported to investigate the security strengths and weaknesses of the deployed MOODLE platform. This might have been due to the fact that many scholars who are proponents of OSS systems rely on the assumption that OSS is designed with the strong security and therefore not vulnerable to attackers and malicious hackers.

The security of the OSS systems is a growing concern for adopters [10]. From its very nature, the fact that MOODLE is made available with open source codes and its development being fully depending on community support which is not officially guaranteed and which constitutes both good guys and bad guys; the MOODLE as the OSS software turns to be of high potential risks from malicious attacks that can be introduced due to systems weaknesses and/or through deliberate and accidental actions by users.
The [15] classified four groups of attacks in MOODLE as authentication, availability (DOS attacks), confidentiality and integrity. The [15] further found that the effective attacks against MOODLE security include username prediction and password prediction by brute-force attack and session attack classified as session hijacking and session fixation.

Despite all the security vulnerabilities of MOODLE LMS and investments, the major concerns of HLIs have been software usability and functionality. None of the adopting organization has taken a lead in assessing MOODLE software security. So it is useful to establish more evidences of security loop holes the MOODLE might undergo as a result of inefficient software customization and management.
Therefore the goal of this study was to familiarize with the current applied methods as being practiced by various HLIs in Tanzania for the adoption and use of MOODLE and thereafter discloses its security vulnerabilities.

III. RESEARCH METHODOLOGY

The research presented in this paper applied a case study methodology. As suggested by [16], this approach has been chosen because security perspective in MOODLE platform can be investigated within its real life context as applied in HLIs with support of multiple sources of evidences.

The **Error! Reference source not found.** lists all stages involved in the study. The first stage was to carry out extensive literature review; then interviews with informants to bring out actual voices. Thereafter, the software was screened at its source codes and configurations so as to establish any security vulnerabilities. Throughout the study, flexibility in asking related issues was maintained to overcome subjective observations. The large amounts of collected verbal information were simple to analyze using empirical material summaries and concepts from grounded theory [17].

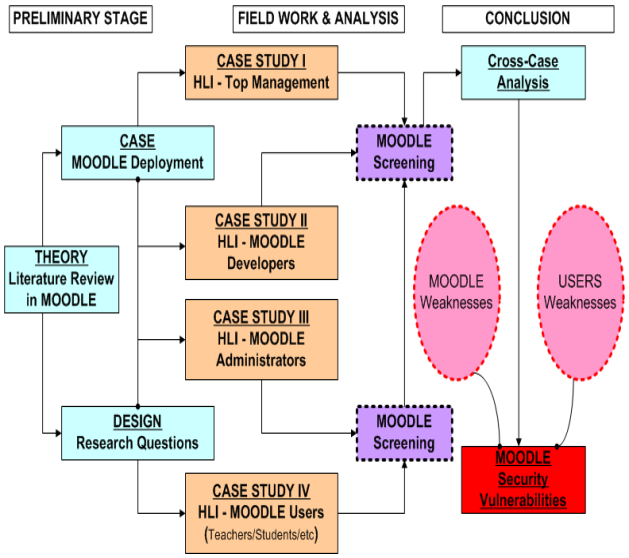

Figure 1: The Research Design

A. Research Settings

Six months of investigation were spent in four HLIs in Tanzania for data gathering. The first three months were spent in HLI-A (September 2015 – November 2015) and the last three months (December 2015 – February 2016) in the remaining HLIs (B, C, and D). So, the period from September 2015 to February 2016 was used for data gathering and system screening. The number of informants involved from each HLI, together with number of registered MOODLE courses and users are presented in the Table 1.

Table 1: Informants, MOODLE Courses & Users

SN	HLI	Informants	Courses	Registered Users	
				Teachers	Students
1	A	29	710	110	12,571
2	B	20	235	43	4,500
3	C	23	157	51	6,327
4	D	15	459	34	1,578
Total		**87**	**1561**	**238**	**30,976**

All four HLIs were selected based on their experience in using MOODLE platform and investment in OSS technology.

B. Data Collection Methods (Sample and Sampling Techniques)

To facilitate the interviewing of the selected respondents, four types of questionnaires were designed to interview *Top Management (TM)*, *MOODLE Developers (MD)*, *MOODLE Administrators (MA)* and *End Users (EU)*. Questionnaires with multiple choice questions, open ended questions and face to face sessions formed the primary data for the research.

In average, each interview session lasted for approximately 30 minutes but interview with MDs and MAs took longer than that because of technical questions and physical investigations of the system. The TMs involves senior level officers in the organization who are responsible for any financial decision regarding MOODLE acquisition and implementation. The MDs are the software engineers who participate in software modification and upgrading. The MAs are responsible for day to day management and administration of the software. Both MDs and MAs have access to source codes. The end-users are the final or ultimate MOODLE users.

In undertaking this study, a total of 87 informants were reached with 4 TMs, 8 MDs, 7 MAs, and 68 EUs. The educational qualifications of the respondents range from senior professors (7), lecturers (19), assistant lecturers (11), and tutorial assistants (7) to non degree and undergraduate students (43).

Table 2 – Academic Profile for Informants

SN	Qualification	TM	MD	MA	EU
1	Senior Professors	4	-	-	3
2	PhD //Lecturers	-	3	-	16
3	MSc / Assistant Lecturers	-	2	4	23
4	BSc / Tutorial Assistants	-	3	2	11
5	Non Degree and Undergraduate Students	-	-	1	15
Total		**4**	**8**	**7**	**68**
Total of **87 informants** interviewed at all sites					

Table 3 – HLIs and Average Years of Respondents

SN	HLI	TM		SD		SA		EU	
		T	A (y)	T	A (y)	T	A (y)	T	A (y)
1	A	1	6	3	6	3	6	24	3
2	B	1	3	2	7	1	4	17	2
3	C	1	2	1	5	2	3	16	2
4	D	1	1	2	4	1	4	11	2
Tt/Avg		**4**	**3.0**	**8**	**2.75**	**7**	**2.4**	**68**	**0.13**

A group of TMs shown to have more experiences in working with the MOODLE with average of 3 years, followed by group of MDs (2.75 years), then MAs (2.4 years) and lastly EU (0.13 years). Most of the end users are students who stay in a university for a specified period of time during their studies.

The visited organizations (HLI – A, HLI – B, HLI – C, and HLI – D) are purposely left anonymous as the consent to disclose identity was refused in order to protect business interests of such organizations. The selection of these HLIs was done through chain and respondent-driven sampling for the purpose of hiding the internal system weaknesses to the public. In each organization, the TMs were approached to allow access to the system and conduct interviews with the staff and students. Then, MOODLE end users (course lecturers and students) were identified and interviewed to get their security understanding and observation. Similar approach was also followed during interview with technical groups of MDs and MAs. In all research settings, data were collected in various ways including unstructured and semi-structured

interviews, documentation analysis, participant and passive observation directly from MOODLE running servers. For theory building study like this, the most important source of information is constituted by interviews [18].

In all HLIs, the software was screened directly from the server by multiple user logins as administrator, teacher, and student. This screening went further to the internal source codes. This has helped researcher to know in deep how the system is configured and the status of default settings and this was a real crucial part of the study.

In the part of documentation analysis, several documents were gathered including the organization's ICT security policies, MOODLE documentation and installation guide [19] and institutional MOODLE training reports.

IV. RESULTS AND DISCUSSION

Generally this study revealed unsecured deployment of MOODLE platform from early software adoption throughout its lifetime. The HLIs are only motivated with user functionalities and put less consideration on its associated security vulnerabilities. The individual MOODLE end-users never feel the security consequences that might happen as a result of poor software customization and configurations of the default settings. Also, technical users (*MOODLE administrators*) perceive that software has natural security just by being an OSS product.

A. The Software Security Coverage in MOODLE Trainings

In this aspect, the interest was to find out if the general issues related to software (MOODLE) security are covered and discussed as part of the MOODLE training sessions. So, the contents of the training packages were evaluated to determine whether they cover all aspects of security and privileges of users in using the system. As part of observation, it has been discovered that the trainings are only targeted to teachers and students for purpose of learning how to create courses, upload contents and accessing materials. This implies that the focus is normally at the functional working of the software and not security.

B. Adoption of Immature MOODLE Platform

Under this item, the interest was to know if the HLIs adopt the latest stable version of the software and if MAs keep track of further MOODLE developments from the trusted sources. This has significant impact in MOODLE security as since software bugs are continually discovered from day to day. The new MOODLE releases fix most of the discovered bugs by default. Unfortunately, it has been found that the task of identifying, selecting and downloading MOODLE software is completely left to MAs who are not sure of the fitness for purpose and interoperability, stability and maturity of adopted MOODLE version.

Table 4 – MOODLE Servers: Versions, Year Installed & OS Type

SN	HLI	MOODLE Version	Year Installed	Operating System
1	A	1.9.19	Feb 2012	Linux - CentOS
2	B	1.7.7	Oct 2010	Linux - UBUNTU
3	C	1.6.9	Sep 2006	Linux - UBUNTU
4	D	1.8.14	Jan 2011	Window Server

The research informs that 50% of the visited HLIs have attempted to download and customize MOODLE software that has not reached functional maturity stage. One of the reasons is that the MAs do not keep track of further MOODLE development from the trusted sources, are not member of online MOODLE community and are not prescribed in the MOODLE mailing lists. In such situation, a move to stable MOODLE version is not guaranteed, and so the HLIs keep investing in the old, outdated and obsolete MOODLE version with highly potential security risks and so become susceptible to malicious attacks.

Table 5 – MOODLE Servers in Use by HLIs: The Versions Status

SN	HLI	MOODLE Version	Support Model	Expiry Date
1	A	2.0.10	Old Version	Dec 2012
2	B	1.7.7	Old Version	Before 2010
3	C	1.6.9	Old Version	Before 2010
4	D	1.8.14	Old Version	Before 2010

The institution A started by installing MOODLE version of 1.6.9 in January 2007.

C. User's Accounts Management in MOODLE

The idea in this aspect was to check whether the accounts requests, creation and issuances follow appropriate channels and if the user passwords are properly managed by active users. It has been clearly observed that there exists a dead link among the software (or *e-learning*) section responsible for MOODLE management, Human Resource (HR) section and the user's department. The user's accounts are given in *unprofessional* way i.e. the requests of the account details are left to individuals. The consequence of this is that; staff continues to own username and password even after retirement or get transferred, change job position or death. This is a very dangerous act for information security in the organization.

Also the trend has shown that most of the HLIs which have more than one web systems with student's records system and mail system being common; users tend to use the same password across different information systems including MOODLE logins.

The easy to guess and weak passwords have been common due to lack of security skills among end users. This create critical danger to the system; for instance, if hacker can guess a weak password created by less skilled user "*teacher*" in a MOODLE platform, hacker can then be able to acquire administrative privileges using the default settings and can do anything of interest in the system. Hackers just need a single

entry point of the MOODLE system to gain administrative power.

D. Security as an add-on Feature and Untrustworthy MOODLE Sources and Support

The security in MOODLE is considered as an *add-on* feature i.e. the software is patched only when the attack happens and bugs discovered. When MOODLE is designed on the first place, the emphasis was given to the systems functionality and usability properties. But since the software depends on the mobile codes which are written by both *good guys* and *bad guys*, it becomes very challenging for MAs to evaluate the source codes so that only clean codes are patched and that cannot introduce new errors in the system. However, the MAs in the HLs are uncertain of the released patches and cannot authenticate the patch sources; they tend to trust the Uniform Resource Locators (URLs), but who is to say a certain website is a reliable and secured source? No habit of checking the veracity of downloads and so there is a possibility of downloading for instance a file named "*login.php*" assuming that it is a MOODLE login file but in reality is a malicious code. Some MAs find their own way for support and solutions of various MOODLE problems through public internet mailing lists. This creates a possibility of revealing organizational confidential information or inheriting support from bad guys that may introduce new security bugs in the software.

E. Organization's Security Considerations in Staff Contracts

Another facet studied was the extent of security consideration in the signed contracts between HLIs and their consulting firms or individuals contracted for MOODLE customization. Several legal documents were screened to establish whether the security requirements are considered or not. It is however revealed that the Terms of References (ToRs) outlined in the legal contracts, Service Level Agreements (SLAs) for software acquisition and support and staff job descriptions lack enough information security responsibilities as far as MOODLE operations are concerned. This might be due to deficient of officers in charge of security in the organization.

F. The MOODLE Code Reviews, System Audit Logs and Data Auditing

The intent of this item was to establish whether or not MOODLE platform and its data are audited from time to time. The monitoring and evaluation of MOODLE source codes and uploaded data is very crucial as MAs have access to the source codes and are so able to modify and change the system behavior. In that possibility, the system auditing is unavoidable; otherwise the HLIs shall only be depending on the ethics and observance of professional codes of conducts by its MAs. The research reports that the HLIs are not sure of the status of the source codes, when and by whom the code changes and whether the updates are documented or not. In most of the HLIs, the MAs have been employed to administrate MOODLE because of their ICT certification and not because of trust. So, the HLIs are uncertain on whether the source codes are manipulated and distorted for illegal purposes or not.

G. The MOODLE Preconfigured Default Settings

Normally MOODLE is received with preconfigured default settings which must also be customized to suit both functional and security requirements. Hackers are aware of these default settings and know how to exploit them if they are left unfixed. The table 6 summarizes observation from the four HLIs as far as customization of the default settings is concerned.

Table 6 – List of Question Items for MOODLE Screening in HLIs

SN	Question Item	A	B	C	D
1	Does MOODLE site registered with ***www.MOODLE.org***?	N	N	N	N
2	Hardcode settings in ***config.php***?	Y	Y	Y	Y
3	Documentation of configuration and code changes	N	N	N	N
4	Are the source code reviews done?	Y	N	N	N
5	Are data & system auditing done?	N	N	N	N
6	Purging (Disable) any unnecessary applications and services in the MOODLE system.	N	N	N	N
7	Handling default accounts with highest privilege level.	Y	N	Y	N
8	Grant access to third parties (Service Providers)	Y	Y	Y	Y
9	MOODLE patches: Are they added?	N	N	N	N
10	Running RootkitRevealer?	N	N	N	N
11	Running SELinux or AppArmor for linux or EMET for windows?	N	N	N	N
12	Teachers can upload script files such as *javascripts*, *flashes*, etc	N	Y	Y	N
13	Ordinary users are allowed to embed Flash and other media in their texts (eg forum posts)	N	Y	Y	N
14	Presence of MOODLE audit logs.	Y	N	Y	N
15	Configure SSL ***httpslogins=yes***	N	N	N	N
16	***register_globals*** (E)nabled or (D)isabled	D	D	E	E
17	Dataroot folder accessible via the web	Y	Y	Y	Y
18	Is your MOODLE open to Google? (Administration Policy)	y	Y	Y	Y
19	Is the password policy activated to force users for stronger passwords?	N	N	N	N
20	Is password salt configured?	N	N	N	N

21	The MySQL network access turned off.	Y	N	N	N
22	Is the Guest access login button visible?	N	N	Y	N
23	Are the User fields hidden?	Y	N	N	N
24	Is there spam protection in Email-based self-registration in MOODLE?	N	N	N	N
25	Can the logged MOODLE guest user be able to execute a Global search?	N	N	Y	N
26	Is it possible to upload CSV files with fields containing quotes?	Y	Y	Y	Y
27	Course creator alters filters at a course level?	Y	Y	Y	Y
28	Awareness of MOODLE Disclosure policy?	N	N	N	N
29	Reported any security issue in MOODLE.org site?	N	N	N	N

After looking in the ***admin/index.php*** in the installation folder to see registration button, the study reveals that MOODLE sites of visited HLIs are not registered with www.MOODLE.org. Among the advantages of registering your MOODLE site is to be able to receive the notifications about recent security issues, patches and updates through email alerts. This also allow the MAs to report any security issue and stay current with PHP, APACHE, MySQL, POSTGRE and MOODLE. Some of the vital mailing lists for MAs to stay updated include CERT [20], PHP [21] and MySQL [22] mailing lists.

The MAs are also required to run regular updates by applying auto update systems such as (Windows Update, Linux: up2date, yum, apt-get, automating updates with a script scheduled via cron and Mac OSX update system). The MOODLE security alerts are also posted online through [29] and RSS feed [30].

In the course of screening the MOODLE source codes and physical checking of the configurations of the MOODLE server, it has been found that two MOODLE systems have their ***register_globals*** enabled despite the fact that this PHP setting requires to be disabled for MOODLE to operate safely. However, the response shows that the 'disabled' status in the remaining HLIs was without knowledge of MAs. Disabling ***register globals*** helps prevent against possible XSS problems in third-party scripts.

In the same line, MAs were not sure of the suitable security status of the MOODLE dataroot folder. The dataroot is the directory where MOODLE stores user files, and this should not be directly accessible via the web. However, the dataroot are left insecure and user files can easily be retrieved publicly through the internet. The user profiles should not be open to the web without authentication, both for privacy reasons and because spammers then have a platform to publish spam on the MOODLE site.

Again in two of HLIs, MOODLE was configured to allow the ordinary users to embed Flash and other media in their texts (eg forum posts). This is a risk action for the security of MOODLE LMS because those rich media objects can be used to steal admin or teacher access, even if the media object is on another server. This is also possible even by enabling flash (.swf) media filter as they can be abused to include malicious flash files.

All HLIs responded that their MOODLE is open to Google. The '*opentogoogle*' setting can be done in *Administration > Security > Site policies.* The MAs were not able to distinguish the use of this facility depending on the e-learning strategy of the organization. The status 'Open to Google' implies that you allow Google to enter your site. This means that all the contents become available to the world. So, how can this be possible while the installed MOODLE is really not a public site?

Regarding the use of the 'password policy', all HLIs informed that the facility is of no use. This implies that activating of the user's password is not centrally controlled. The password policy is significant for enhancement of MOODLE security as it forces user to use stronger password that are less susceptible to being cracked by intruder. Again as with 'open to Google', the MAs can enforce password complexity by check box from *Settings > Site administration > Security > Site policies*. This facility provides the option to set the minimum length of the password, the minimum number of digits, the minimum number of lowercase characters, the minimum number of uppercase characters and the minimum number of non alphanumeric characters. If user enters a password that does not meet those requirements, they are given an error message indicating the nature of the problem with the entered password. Enforcing password complexity along with requiring users to change their initial password goes a long way in helping ensure that users choose and are in fact using strong passwords.

One of the very crucial security features of MOODLE platform is the use of the '***password salt***'. The MAs from all visited HLIs were found to be ignorant of the uses and advantages of this facility, and thus all MOODLE systems have been left with default password salt. Setting a password salt greatly reduces the risk of password theft. One of the advantages of using password salt is to avoid a possible decrypt of MOODLE password using MD5 hash algorithm [23]. The MD5 hash algorithm can be broken [24]. The MD5 algorithm stands for **Message Digest** algorithm **5,** a widely used cryptographic hash function that takes up a random data (text or binary) as an input and generate a fixed size 'hash value' as the output. A random string of characters is added to passwords before their MD5 hash is calculated, a process

which makes them harder to reverse (i.e. the longer the random string, the harder you make it). The MOODLE Salt Generator is normally used to obtain a suitable long (>= 40 characters is recommended) random string. However, it is recommended that the password salt should be stored in another safe place other than ***config.php*** to prevent loss and allow future password recovery. In a case of changing the password salt (including importing users from another MOODLE site that uses password salt), all old salts (up to 20) must be retained in ***config.php*** in addition to the new salt until every user has logged in at least once, so they start using the new salt.

A password salt can be set by adding the piece of programming line to your ***config.php*** file such as: **"$CFG->passwordsaltmain = 'long random string with many characters';"**.

Additionally, the password salt needs to be changed regularly.

The response regarding the hardcode settings in the ***config.php*** file was 'yes' from all HLIs and the PHP file is set to be writeable by the web server process, the action that compromise MOODLE security. For a writable ***config.php*** file via a web server process, then it is possible for another vulnerability to allow attackers to rewrite the MOODLE code and display whatever they want. Displaying of PHP errors is another possible security issue as this can allow anyone to enter a faulty URL causing PHP to give up valuable information about directory structures.

All HLIs responded that their MOODLE is no configured with Secure Socket Layer (SSL). Although MOODLE does not support the SSL implementation all over the site [25], but however, this can be looked as one of the major requirements for software improvement. The SSL is the standard security technology for establishing an encrypted link between web server and browser. The SSL for the entire site (connection with clients) should be applied to prevent session fixation, session hijacking and username prediction by adding a PHP scripts that change the content of four variables of CFG as follows**: themewww** (*change http to https*), **wwwroot** (*change http to https*), **loginhttps** (*encrypted using SSL and script turned on*), **httpstheme** (*change http to https after loginhttps is on*) [26]. The global flag is set to true i.e. (HTTPSPAGEREQUIRED = true). Again to protect session fixation for the first login for the HTTP protocol without SSL, a new ID session should be generated. Normally this can be done by modifying *moodlelib.php* file stored in MOODLE lib package.

H. The MOODLE Security Vulnerabilities

The most common observed security vulnerabilities for MOODLE software include Cross-Site Scripting (XSS), security bypass, SQL Injection, *Cross-Site Request Forgery (*CSRF) and PHP Code Injection, Authorization Vulnerabilities and Arbitrary File Upload. These vulnerabilities are well discussed in [27].

H-1. Cross Site Scripting (XSS) Security Vulnerabilities

In some HLIs, the MNET access control interface in MOODLE has a persistent XSS vulnerability which happens when server allows extended characters in usernames. This system has also XSS vulnerability in the file **blog/index.php** where some parameters are not being properly cleaned on the blog index page, allowing non-persistent XSS attacks. In the global search engine, MOODLE has a reflexive XSS which is a problem in handling of user submitted data in global search forms. This problem is exploitable only when global search is enabled. However, by default the global search feature is disabled. When using a *login-as* feature in MOODLE, users may trick admins to edit some existing posts which contain XSS exploit code.

H-2. Security Bypass Vulnerabilities

MOODLE has KSES (*Kses Strips Evil Scripts*-"HTML Filtering Mechanism") Security Filter Bypassing vulnerability. This is a critical vulnerability in KSES text cleaning filter which may allow registered users to launch persistent XSS attacks.

H-3. SQL Injection Vulnerabilities

MOODLE data may be passed through **add_to_log()** function without being sanitized properly, this could allow SQL injection type attacks if there are any instances of wiki in the courses.

H-4. The CSRF and PHP Code Injection Vulnerabilities

The Cross Site Request Forgery (CSRF) is a possibility of an attacker to add new administrator in the system. The remote attacker can trick an administrator to visit a malicious site; the attacker can perform privileged operations, or exploit PHP code injection vulnerability. The successful exploitation of these vulnerabilities can lead to arbitrary code execution. The CSRF happens in MOODLE **Quiz reports** where only limited validation is being done for one of the parameters, allowing unauthorized deletion of attempts in some instances.

V. CONCLUSION AND FUTURE WORK

The study reveals evidences on how HLIs in Tanzania are exposed to security attack of their MOODLE systems as a result of poor software customization and an *ad hoc* adoption process from early stages for software selection, installation, usage and upgrading. There is a technical gap existing between MDs and the local MAs. The MAs do not keep track of further MOODLE developments and most of the sites are not registered in www.moodle.org. The MAs are not aware of MOODLE security issues and have never reported any

incidence in the main MOODLE site. This means that these HLIs are operating in an isolated way from both MOODLE and software security developments.

Despite the fact that the rate of investing in MOODLE platforms is increasing in Tanzania, no security measures are seriously taken for protection. Any security misconduct may lead to the failure of organization from attaining intended benefits. The HLIs are advised to observe the COBIT based framework for the secure adoption and use of open source information systems [28].

There is need to conduct a research for other web based open source information systems with similar features like MOODLE so as to establish a general trend of security vulnerabilities originated from software design and customization, poor configurations, and adoption procedures.

VI. ACKNOWLEDGEMENT

I would like to express my sincere gratitude and many thanks to the management of the Open University of Tanzania for financing this research through a small research grants fund of the Directorate of Research and Postgraduate Studies.

.

REFERENCES

[1]. OSI (2004) The Open Source Definition v1.9 online: http://www.opensource.org/docs/definition.php Accessed March 12, 2016.

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

[3]. S. Shearer, "Open Source Software in Education", *TheCompton School: London*, 2003

[4]. http://www.Moodle.org. Accessed January 05, 2016.

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

[6]. "Moodle stats page" from www.Moodle.org/stats Accessed March 20, 2016

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

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

[9]. Weerawarana, S., & Weeratunga, J. (2004). Open Source in Developing Countries. Stockholm: Edita Sverige AB.

[10]. Schneider, F. B., (2000), "*Open Source in Security: Visiting the Bizarre*." Proceedings of the 2000 IEEE Symposium on Security and Privacy (the Oakland Conference), Berkeley, CA. Los Alamitos, CA: IEEE Computer Society. pp. 126-127.

[11]. Bakari, J. K., (2007), "*A Holistic Approach for Managing ICT Security* in *Non Commercial Organizations*", A Case Study in a Developing Country; PhD Thesis, Stockholm University, Sweden, ISBN - 91-7155-383-8.

[12]. Steven Weber, "Open Source Software in Developing Economies", http://www.ssrc.org/programs/itic/publications/ITST_materials/webernote2.pdf Accessed November 2015

[13]. Lungo J. H. and Kaasbøl (2006), "*Experiences of open source software in institutions: Cases from Tanzania and Norway*".

[14]. Mitchell, C., (2004), "*Understand your open source software options*", http://www.cioupdate.com/trends/article.php/3419381, Accessed: February 23, 2016

[15]. Vahe A. Arakelyan, (2013), "Vulnerable Security Problems in Learning Management System (LMS) Moodle", Mathematical Problems of Computer Science 39, 129—134, 2013

[16]. Yin, R., (1989), "Case Study Research". Sage Publication, California, pp: 22-26.

[17]. Glaser, B. G., & Strauss, A. L. (1967). The Discovery of Grounded Theory: Strategies for Qualitative Research. New York: Aldine de

[18]. Walsham, G. (1995) The Emergence of Interpretivism in IS Research, Information Systems Research 6(4): 376-394.

[19]. MOODLE Documentation and Installation Guide, Accessed August 2015

[20]. CERT mailing list http://www.us-cert.gov/cas/signup.html Accessed March 24, 2016

[21]. PHP mailing list http://www.php.net/mailing-lists.php Accessed March 24, 2016

[22]. MySQL mailing list http://lists.mysql.com Accessed March 24, 2016

[23]. https://en.wikipedia.org/wiki/**MD5**, *Accessed March 16th 2016*

[24]. X. Wang, D. Feng, X. Lai and H. Yu, "Collisions for hash functions MD4, MD5, HAVAL128 and RIPENMD", in Cryptology ePrint Archive, http://eprint.iacr.org, (Accessed 25 January 2012), Report 2004/199, 2004.

[25]. Sheo Kumar & Kamlesh Dutta, (2011), "Investigation on Security in LMS MOODLE", International Journal of Information Technology and Knoweldge Management January-June 2011, Volume 4 No. 1 pp 233-238

[26]. J. Carlos, G. Hernández and C. M. A. León, "Moodle security vulnerabilities", *5th International Conference on Electrical Engineering, Computing Science and Automatic Control (CCE)*, ISBN: 978-1-4244-2499-3, 2008

[27]. Ally, S. (2014), "Security Vulnerabilities of the Web Based Open Source Information Systems: Adoption Process and Source Codes Screening". HURIA Journal, Vol No: 17, ISSN: 0856 6739, Nov 2014

[28]. Ally, S. (2011), "Secure Adoption and Use of Open Source Information Systems in Tanzanian Organizations", MSc Dissertation, Open University of Tanzania, 2011.

[29]. MOODLE Security Mailing list on Web, http://MOODLE.org/security, Accessed March 21, 2016

[30]. RSS feed, http://MOODLE.org/rss/, Accessed March 21, 2016

Author Profile

S. Ally received the **B.Sc.** degree in **Computer Science** from the University of Dar Es Salaam, Tanzania in 2006 and **M.Sc.** in **Computer Science** from the Open University of Tanzania in 2011. Currently, has applied for a **PhD** programme. His research interest includes Hypervisors, OSS security and software engineering, science data management, mathematical modeling with Fast Fourier Transform (FFT), Astro-informatics & Virtual Observatory (VO), Compiler Technology, Artificial Intelligence & Expert Systems and e-learning technologies.