2018 International Conference on Computational and Characterization Techniques in Engineering & Sciences (CCTES)
Integral University, Lucknow, India, Sep 14-15, 2018

E-learning Platforms Security Issues and Vulnerability Analysis

Meghna Bhatia
Assistant Professor
SIES College of Arts, Science and Commerce
Navi Mumbai, India
meghnabhatia@rediffmail.com

Dr. J. K. Maitra
Associate Professor
Department of Maths and CS
RDVV University,
Jabalpur, India

***Abstract*— E-Learning is a fairly a new concept of learning electronically, mostly through the internet. Web applications and E-learning Platforms are important and billions of users rely for performing routine activities. User-friendliness, 24/7 availability and ubiquitous of web applications have made them vulnerable. A unpretentious implementation fault in the application could allow an invader to steal sensitive information and perform adversary actions, and hence it is important to secure web applications from attacks. Defensive mechanisms for securing web applications from the flaws have received attention from both academia and industry. In this paper , all available modern-day open-source learning management systems (E-Learning software) are evaluated for security issues and vulnerabilities. Those platforms were thoroughly tested using two web vulnerability scanners due to the big variances among security scanners. The key aim was to assess this software category illustrating valuable conclusions on their security level today, since such popular categories of software should above all ensure that security is a must, something that seems to be neglected: until now this category of software was not tested in the literature for weaknesses; The major contribution of this paper is to inform the educational community about the prevailing vulnerabilities and the probable risks and flaws with the significant results, and offer a valuable contribution to the developing communities of the above platforms by recommending a significant security model. This security model can be easily adapted to the needs of every different e-learning platform or even web platforms in over-all since this is an almost non-existing topic in the current scientific literature.**

Keywords— Learning Management System (LMS) , Vulnerability , Open Source Software, e-learning Platform ,Authentication plug-ins .

I. INTRODUCTION

Having added more and more organizations offering online courses, or even full study programs, information security tends to become important issue in education too (Chen Y and He W ,2013). According to a study carried out by Violettas, Theodorou, & Stephanides, 2013 the authors tested four open-source e-learning software suites for vulnerabilities. The results were disappointing since several weaknesses were found in the four suites examined. In this work all available contemporary open-source e-learning platforms are evaluated for security vulnerabilities and flaws. These e-learning platforms are tested using two vulnerability scanners: Netsparker Community Edition, (Netsparker Community Edition, 2014) and N-Stalker (N-Stalker, 2014). The results are discussed and compared against today's lists of vulnerabilities defined by widely recognized security institutes. There is also a comprehensive research in the literature about the evaluation of the e-learning software platforms mentioned below (e.g. (Attwell, 2006), (Liaw, 2008), (Shee & Wang, 2008), (Kurilovas, 2009), (Hussain, Wang, & Sun, 2011) and their related basic features and capabilities.

On 2010, 97.5% of all institutions of higher education in the United States have deployed at least one e-learning platform or LMS (Williams van Rooij, 2012). This type of software has some common characteristics irrespective of whether it is open-source or commercial, installed on a local server or provided as a service. All the software above serves the same purpose, in approximately the same way: material is divided into modules and lessons with different levels of user's rights depending on the user's role, i.e. student, teacher, and administrator. These users are further classified based on their individual role, year of study, enrolled lessons, and assigned exercises. Also, many e-learning suites offer free access to a portion of the suite to (sometimes anonymous) users not affiliated with the institutions running this software. All this complicated hierarchy, along with the complexity and differentiation of the teaching material and purpose (video, audio, PDF, or even executable programs) must be organized, secured and well served in an environment of academic freedom where research, curiosity and even pranks are allowed and encouraged. Despite all this, the security aspect of this software category is not extensively studied in the literature; this leaves all the stakeholders (teachers, students, administrators, developers) without a referencing guide and theoretical guidance.

Some have been asking to deal with the security of e-learning as a distinguished part of Information Security, making the current work absolutely essential: "…Information Security risks are not necessarily unique to the e-learning environment but should however be addressed as if it were. It is therefore vital that all necessary steps be taken by educational institutions to ensure information is properly secured within the e-learning environment…" (Kritzinger & Von Solms, 2006, p. 1).

On the other hand, there are many works that don't consider the security aspect at all. In a guide for the European Commission on e-learning evaluation (Attwell, 2006), the security aspect is not mentioned anywhere. There are several questions about software usage and user concerns, but the most important user (i.e. student) concern, i.e. personal data treatment (May, Fessakis, Dimitracopoulou, & George, 2012), is addressed nowhere.

Kurilovas (Kurilovas, 2009) evaluated several e-learning software packages. The author examined the multiple

evaluation criteria of Learning Object Repositories (LORs) and their technological quality based on score ranking results.

Liaw (Liaw, 2008) conducted a detailed survey on the user's perceived satisfaction using e-learning, on a case study by using the "Blackboard" e-learning platform. According to the author, there are four elements that should be considered when developing e-learning environments: "environmental characteristics and satisfaction", "learning activities" and "learner's characteristics". In this survey, there are several questions but not on any aspect of security issues.

Shee & Wang (Shee & Wang, 2008) described a multi-criteria evaluation of the Web-Based e-Learning System(s) (WELS), by defining four different dimensions (groups) of evaluation criteria; the word "security" is absent and it could only be implied under "C04 Operational Stability".

In the paper contrary to the above, it is argued that security should be addressed as a necessary criterion of a software product.

II. E-Learning Security

The usage, the potential, and the effectiveness of e-learning are all topics very well discussed in the literature today. On the contrary, the security aspect of e-learning has not received much attention today (Jeghal, Oughdir, & Tairi, 2014) resulting in very poor literature findings on the matter.

(El-Khatib, Korba, Xu, & Yee, 2003) examined "…the more popular e-learning standards to determine their provisions and limitations for privacy and security…" (El-Khatib et al., 2003, p. 1). They clearly state that "… Most e-learning innovations have focused on course development and delivery, with little or no consideration to privacy and security as required elements…" (El-Khatib et al., 2003, p. 2) acknowledging the fact of the lack of security aspect on such software. The roles of security in eLearning software include user authentication/authorization, protection of private information from unintended access, and protection of data integrity.

(May & Sébastien, 2011) claimed that security and privacy problems in eLearning environments are increased, leading to a situation where security and privacy are becoming essential to the users. They state that in an environment such as an eLearning suite, the exchange of personal and collaborative data is intense, so a strong protection of participant's privacy should be an absolute must. In the same essay, we find that users (i.e. students) are highly concerned about their privacy, the use (and misuse) of their personal data and the possible hijacking. The authors are willing to raise awareness on neglecting the implications of student tracking and use of their personal data in the research efforts. "…The crucial tasks for learning service and content providers are to secure learning environment and to secure storage of learner data…" (May & Sébastien, 2011, p. 5). According to the study, awareness raising, protection of personal data, the authenticity of learning resources, seamless access, address and location privacy, single sign-on, digital rights management, legislation and anonymous usage (May & Sébastien, 2011, p. 5) are all important factors of e-learning systems usage.

(Huang, Lin, & Huang, 2012) proved that when a student "…possessed relatively higher prior knowledge [on the subject taught in the e-learning environment] the passive participation had a strong and positive effect on e-learning performance…" and "…passive participation is positively related to e-learning performance…" (Huang et al., 2012, p. 346). Consequently, the authors conclude that sensory students demonstrate a higher level of online participation. Hence, those students may not be the most vulnerable to security flaws, but they are definitely the prime suspects for misusing the knowledge they acquired through their participation in the e-learning platform.

(Weippl & Ebner, 2008) are using the "classical" CIA (Confidentiality, Integrity, and Availability) approach for security, claiming that all other requirements can be traced back to those three basic properties. Extending those with other "difficult to fit" categories, such as non-repudiation or accountability, is an open discussion today (Whitman, 2012), (ISO/IEC 13335-1, 2004).

(Luminita, 2011) is stating that the six basic security aspects (authenticity, access control, confidentiality, integrity, availability, non-repudiation) should be met for e-learning platforms. The author also cites all the most common vulnerabilities affecting e-learning software including SQL injection (SQLi), Cross Site Scripting (XSS), password cracking, hijacking and guessing session id.

III. Methodology of the Study

In this section, important information regarding methodological issues of the study is presented, namely: the software vulnerability taxonomies utilized; the vulnerability scanners used, and the open-source e-learning platforms selected for vulnerability evaluation. As already mentioned, the paper is aiming at opening the discussion on how to conduct a security evaluation on an application or group of similar applications and the guide to follow. The goal of this article was to evaluate the E-learning group platforms, so every effort was put to gather all such software available today as described in section 3.3.

A. (Web) Software Vulnerabilities Taxonomies

Web application security is a difficult issue since -by definition- such applications are exposed to malicious users. The input to those applications is mainly generated within HTTP requests, hence this input has to be sanitized; most of the vulnerabilities are exploited by the poor or non-sanitization of this input. This is not because there are no available programming tools and techniques today, but rather because of the absence of input validation (Scholte, Balzarotti, & Kirda, 2012). Other types of weaknesses include poor authentication mechanisms, logical weaknesses, unintentional disclosure of content and information and low-level coding weaknesses such as buffer overflow (Fong & Okun, 2007) and the notorious XSS and SQLi where the "legacy problem" seems to have found another major "application field" (MITRE, 2011b) to less popular applications (Scholte et al., 2012).

We used the most widely used security taxonomies as presented right below, in order to classify the weaknesses

found in the e-learning software examined. Look at TABLES VI & VII, where every weakness found (when possible) is mapped to one or more of the following classifications.

B. CWE (https://cwe.mitre.org/)

The Common Weakness Scoring System (CWSS) is a collaborative, community-based effort that is addressing the needs of its stakeholders across government, academia, and industry…" (Martin, 2014, p. 1). The scoring of CWSS is depicted in the second column of TABLE I.

The 2011 CWE/SANS Top 25 (Table II) Most Dangerous Software Errors (MITRE, 2011a) is a list of the most widespread and critical errors/vulnerabilities that lead to serious vulnerabilities of software. It attempted to perform quantitative prioritization of CWE entries using a combination of Prevalence and Importance.

TABLE I. CWE/SANS TOP 25 MOST DANGEROUS SOFTWARE ERRORS

CWE Rank	CWSS Scoring	ID	Subhead
1	93.8	CWE-89	SQL Injection
2	83.3	CWE-78	OS Command Injection
3	79	CWE-120	Classic Buffer Overflow
4	77.7	CWE-79	Cross-site Scripting
5	76.9	CWE-306	Missing Authentication for Critical Function
6	76.8	CWE-862	Missing Authorization
7	75	CWE-798	Use of Hard-coded Credentials
8	75	CWE-311	Missing Encryption of Sensitive Data
9	74	CWE-434	Uploads of File(s) of Dangerous Type
10	73.8	CWE-807	Reliance on Untrusted Inputs in a Security Decision
11	73.1	CWE-250	Execution with Unnecessary Privileges
12	70.1	CWE-352	Cross-Site Request Forgery (CSRF)
13	69.3	CWE-22	Path Traversal
14	68.5	CWE-494	Download of Code Without Integrity Check
15	67.8	CWE-863	Incorrect Authorization
16	66	CWE-829	Inclusion of Functionality from Untrusted Control Sphere
17	65.5	CWE-732	Incorrect Permission Assignment for Critical Resource
18	64.6	CWE-676	Use of Potentially Dangerous Function
19	64.1	CWE-327	Use of a Broken or Risky Cryptographic Algorithm
20	62.4	CWE-131	Incorrect Calculation of Buffer Size
21	61.5	CWE-307	Improper Restriction of Excessive Authentication Attempts
22	61.1	CWE-601	Open Redirect
23	61	CWE-134	Uncontrolled Format String
24	60.3	CWE-190	Integer Overflow or Wraparound
25	59.9	CWE-759	Use of a One-Way Hash without a Salt

C. OWASP (https://www.owasp.org)

The OWASP Top Ten (TABLE II) covers more general concepts than CWE and is focused on web applications. On the other hand, the CWE Top 25 covers a broader range of issues than what arises from the web-centric view of the OWASP Top Ten. There is some overlap, however, since web applications are so prevalent, and some issues in the OWASP Top Ten have general applications to all software classes (OWASP, 2013).

TABLE II. OWASP TOP TEN 2013 VULNERABILITIES

Classif-ication	Name	Description
A1	Injection	Injection flaws, such as SQL, OS, and LDAP injection
A2	Broken Authentication and Session Management	Application functions related to authentication and session management incorrectly implemented, leading to compromise of passwords, keys, or session tokens, or to exploit other implementation flaws to assume other users' identities
A3	Cross-Site Scripting	Taking untrusted data and sending them it to a web browser without proper validation or escaping. XSS allows attackers to execute scripts in the victim's browser which can hijack user sessions, deface web sites, or redirect the user to malicious sites
A4	Insecure Direct Object References	Exposing a reference to an internal implementation object, such as a file, directory, or database key. Without an access control check or other protection, attackers can manipulate these references to access unauthorized data.
A5	Security Misconfiguration	Secure configuration defined and deployed for the application, frameworks, application server, web server, database server, and platform. Secure settings should be defined, implemented, and maintained, as defaults are often insecure. Software should be kept up to date.
A6	Sensitive Data Exposure	Not properly protecting sensitive data, such as credit cards, tax IDs, and authentication credentials. Sensitive data deserves extra protection such as encryption at rest or in transit.
A7	Missing Function Level Access Control	Verifying function level access rights before making that functionality visible in the UI. However, applications need to perform the same access control checks on the server when each function is accessed. If requests are not verified, access to functionalities without proper authorization is possible.
A8	Cross-Site Request Forgery (CSRF)	A CSRF attack forces a logged-on victim's browser to send a forged HTTP request, including the victim's session cookie and any other automatically included authentication information, to a vulnerable web application. This makes the victim's browser to generate requests the vulnerable application thinks are legitimate requests from the victim
A9	Using Components with Known Vulnerabilities	Components, such as libraries, frameworks, and other software modules, almost always run with full privileges. If a vulnerable component is exploited, such an attack can facilitate serious data loss or server takeover. Applications using components with known vulnerabilities may undermine application defenses and enable a range of possible attacks and impacts.
A10	Unvalidated Redirects and Forwards	Web applications frequently redirect and forward users to other pages and websites, and use untrusted data to determine the destination pages. Without proper validation, victims can be redirected to phishing or malware sites, or use forwards to access unauthorized pages

D. CAPEC (http://capec.mitre.org/)

While CWE is a list of software weakness types, CAPEC is a list of the most common methods used by attackers to exploit the vulnerabilities described in CWE. In other words, CWE just describes the weakness; whereas CAPEC describes the ways to use the weakness for a successful attack.

E. WASC Thread Classification

The WASC Threat Classification is a cooperative effort to clarify and organize and enumerate the attacks, and weaknesses that can lead to the compromise of a website. It provides a proper classification of threads into attacks and weaknesses. Its main usage is to provide the industry with authoritative material and support which has been peer viewed for maximum possible accuracy. It is not as comprehensive as the CWE list, targeting the masses of "active" industry (developers, Security Officers, etc.) and not the Academia as CWE does.

F. Vulnerability Scanners

Today there are more than 120 web vulnerability scanners approved by the Security Standards Council (PCI, 2017), even though some of the top names and brands on the market are not included in the above Security Standards Council list.

No individual scanner today is a top-performer in terms of vulnerability discovery. Nevertheless, scanners do, in general, expand the testing effort in rough proportion to the vulnerability population in the wild, i.e., information leakage vulnerabilities, XSS and SQLi (Bau, Bursztein, Gupta, & Mitchell, 2010).

Two well-known scanners were chosen to conduct the research, Netsparker (Netsparker Community Edition, 2014) and Acunetix (Acunetix, 2014). They were both chosen for the following reasons: They are user-friendly

1) They both provide an explanation(s) about the vulnerabilities found
2) They are considerably fast and they don't require any specific setup. They do not have big demands in hardware (RAM, CPU, etc.)
3) They are very well known in the market and they scored very high in extensive tests (Shay, 2014)
4) They both provide free editions:
   a) NetSparker provides the Community Edition which was used. The only drawback of this edition is that it does not reveal possible solutions for the vulnerability found as the professional edition does. Since this is a survey, this functionality was not a concern
   b) Acunetix is full edition but limited for 30 days of use. Since the survey was done in a shorter time period, this was again not an issue

Both scanners were tested against each one of the OWASP top ten 2013 vulnerability categories.

G. Open-Source E-Learning Platforms Chosen

From the dozens of LMS platforms existing today the selection of the platforms examined in this study was based on the following criteria:

1) *Open-source:* In open-source software, it is possible to access the source code and is much easier to conduct a study since no financial or copyright problems exist. Open-source code LMS has the biggest share on the market today and is very preferable by the academic community. Moodle alone, for example, has a wide base of installations on thousands of institutions (Williams van Rooij, 2012).
2) *Extent of usage:* The number of students/teachers using the platform today was also a very important criterion. There are many cases of LMS that looked very promising but they either deteriorated due to financial issues or merged with other similar projects. The LMSs selected in this paper are active today and they support hundreds of thousands of students.

In this work, we examined all of the open-source e-learning software available in the market today, to test their security level. In TABLE III there are all the e-learning platforms that were tested in the paper.

TABLE III. E-LEARNING PLATFORMS EXAMINED

#	E-Learning Platform
1	Dokeos (http://www.dokeos.com/)
2	ILIAS (http://www.ilias.de/)
3	ATutor (http://atutor.ca/atutor/)
4	Sakai (https://sakaiproject.org/)
5	Moodle (https://moodle.org/)
6	Claroline (http://www.claroline.net/)
7	Docebo (http://www.docebo.com)
8	eFront (Educational) (http://www.efrontlearning.net/)
9	OLAT (http://www.olat.org/)
10	Chamilo LMS (http://www.chamilo.org/)
11	Canvas (http://www.instructure.com/)
12	LAMS (http://www.lamsinternational.com/)
13	LON-CAPA
14	Commsy
15	Fedena

IV. RESULTS

It is important to state here the difference between declaring (stating) that a given software has a particular weakness, and state that this given software has a weakness that is classified as severe, it is well documented that it leads to those certain attacks, it can be exploited in such ways, hence it has to be addressed (fixed) in a proper manner. The latter is not given by the scanners and was done by the authors, so the theoretical background of those vulnerabilities found can be discussed among stakeholders. Moreover, such a systematic research can lead to a better understanding of those flaw and potentially discover new ways and means to confront them.

A. Classifications of Weaknesses

It is obvious that different vulnerability scanners use completely different and incompatible lists and taxonomies of vulnerabilities. Any attempt to homogenize the results of those different scanners in order to present them in a uniform way yields unacceptable results: this is because different vulnerabilities will not only be "squeezed" under a common label but rather because there is a big difference between the vulnerabilities found by different scanners.

"Acunetix" is far more up-to-date, searching for vulnerabilities not yet documented into the literature such as BREACH (Gluck, Harris, & PRADO, 2013) or specific XSS vulnerabilities of particular software parts versions.

For the reasons above, vulnerabilities found by each scanner were presented into separate tables with the corresponding description and impact. All vulnerabilities found were classified based on the taxonomies described in section 3.1

In the classification of vulnerabilities found by "Netsparker" (TABLE IV) the vulnerability named "Password Transmitted over HTTP" is obviously included in "1Basic Authorization over HTTP"; various flavors of this same vulnerability are mentioned separately only for educational purposes.

In the taxonomy of vulnerabilities found by "Acunetix" (TABLE V), they were grouped where possible under the same "family". The correspondence to OWASP-2013, CWE/SANS. CAPEC, WASC taxonomies is not given by the scanner, so it was done by the authors. In a few cases where the vulnerability did not match with the corresponding taxonomy, the field was intentionally left blank.

B. *Synopsis of Findings*

In Fig. 1, the percentages of the severe findings of both scanners are presented. This means that if the vulnerability was found by both a scanner is counted once, but it is also counted if found by only one of the scanners on a specific platform. Obviously, not all vulnerabilities are common for both canners, so where applicable, one "super" category was formed from related vulnerabilities (e.g. SQLi(s), XSS of various forms). Although BREACH and CRLF vulnerabilities are a kind of XSS attack, they were not included under XSS just to point out their severity

In Fig. 2, the same "super" category logic as in Fig. 1 applies as well. The server-side vulnerabilities presented is not included because as it is already mentioned they don't belong to the "duties" of the application. Finally, in Fig. 3, total vulnerabilities for all platforms from both scanners are depicted. Obviously, those vulnerabilities the scanners have "in common" are calculated twice (e.g. XSS is reported once for each scanner for the same application).

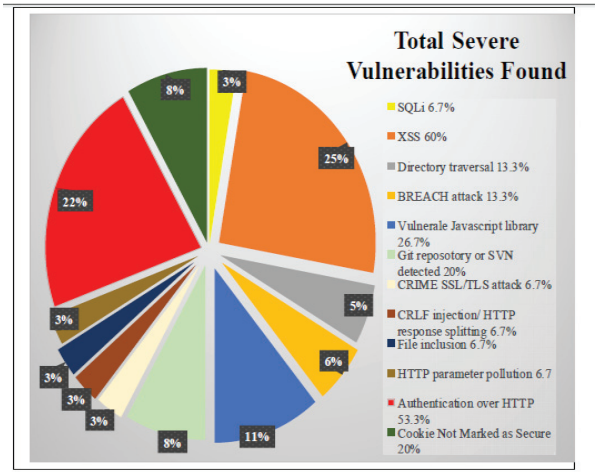

Fig. 1. Severe Vulnerabilities percentages found by both scanners over all platforms.

We need to emphasize here that the percentage of the total vulnerabilities found by vulnerability scanners, in general, is something very uncertain; different scanners detect different types of faults. In (Bau et al., 2010) a group of scanners is evaluated, with "Acunetix" among them. The authors created custom SQLi vulnerabilities and tested them against all scanners under examination. The detecting score was no more than 15%! In (Fonseca, Vieira, & Madeira, 2007), two of the scanners tested discovered SQLi in no more than 20% with a very high number of false positives which obviously diminishes the confidence of the users.

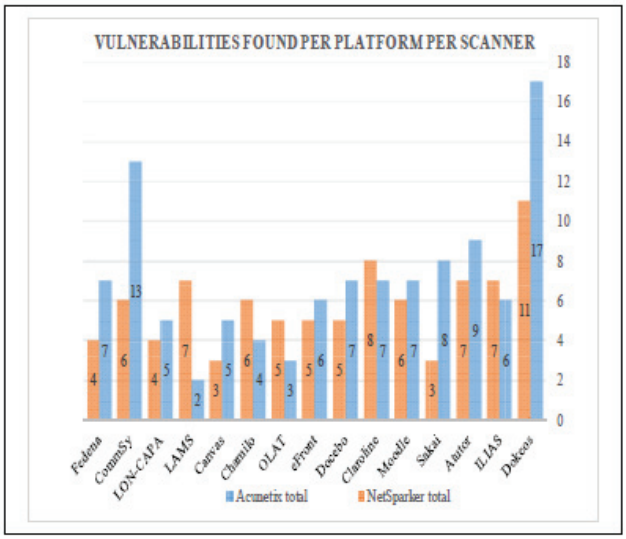

Fig. 2. Severe Vulnerabilities total number found by both scanners per platform.

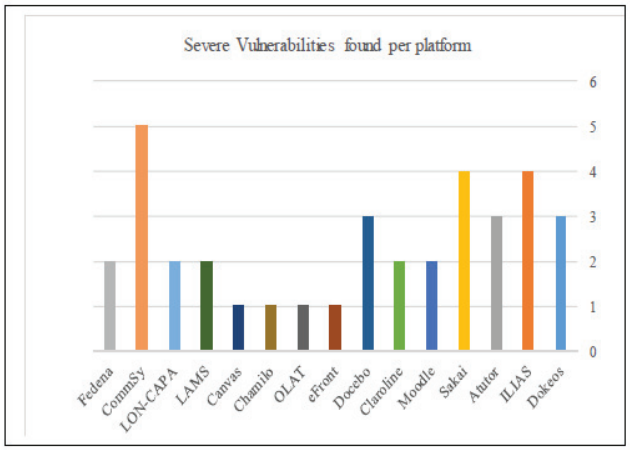

Fig. 3. Total vulnerabilities found by both scanners for all platforms.

V. Proposed Model For Ensuring Security In E-Learning Platforms

Based on the findings, with the discussion of the immense breach of security present in the existing Learning Management Systems and E-learning platforms, the authors pose an abstract vision of their recommended security model as a part of sheathing the learning management system with a security cover. Primarily the security model is posed with a two-fold holistic view.

A. Hierarchical Approach

A normalized learning management system is mostly structured in some hierarchical fashion. The hierarchy holds the system administrator at the top-tier of the control-structure tree. It then permeates to the instructors and then to the pool of learners. The access to the study resources is restricted and controlled primarily by the sole administrator and then by the instructors. To enforce the security sheath in a top-down fashion, the security flow initiates with the administrator of the system.

The chief advantage of such a system is that the security monitoring could be done in a centralized manner and enforce security from the topmost level would definitely cover the entire system within the security sheath. However, there are issues to implement such a system in scenarios where there is a constant change in the hierarchical structure. In such cases, every slightest change in the design would raise the need to reshuffle the security model. Also in a large-scaled system having an immensely huge number of users, large overhead is incurred to include every new user within the entire security structure. However, this model is expected to work well in properly defined and structured eLearning systems with a moderately defined number of users.

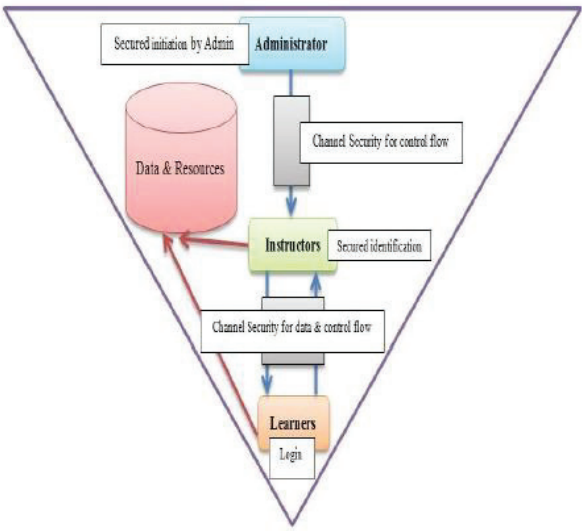

Fig. 4. A top-down approach for enforcing security in eLearning systems

B. Distributed Approach

Just on contrary to the centralized approach in enforcing security over the eLearning systems, distributed approach poses a separate structure. Distributed structure keeps separate

security models for each of the elements in the eLearning system. The security models may follow different security logics but a continuous interaction is needed between the peers. The chief advantage of this system is that scalability never turns to be a bottleneck. With the inclusion of components, the entire security model could be exempted from a total modification. On one hand this opens the scope of segmented modification of the system and on the other hand, adds flexibility in choosing the security mechanisms.

To encapsulate the entire system within the security cover, the primary task is to highlight the areas of vulnerabilities and their effects. The goal of a potential attacker could be multifarious.

The primary target could be the course contents and the private areas of the system (evaluation, certification, personal details of the learners etc.).

The second type of attack could be masquerading as students which could also be serious from the examination point of view. Denial of service could also be another serious aspect of a potential attack. The entire security model could be structured based on a timeline as follows:

- The first objective would be framing of the security policies for the eLearning system. This phase would deal with the feasibility study and design of the mechanisms efficient to ensure the security of the eLearning model.
- The second phase would definitely cover the implementation of the security models within the eLearning system.
- The third phase would employ a constant monitoring. Detection of an attack (or even an eavesdropping) at the earliest could be the best way to combat it. And this phase would constantly monitor the system for potential breach of security or any unwanted access.
- The fourth phase could be termed as the "*SOS phase*" which needs to be invoked in case an attack has been detected. The primary task of this phase is twofold;
  - to safeguard a maximum possible portion of the system from any further attack and
  - to set right the damaged things along with stringent actions against the attacker.

Out of all the four phases of the security management system in eLearning, the third phase needs to be constantly running which would invoke the fourth phase on demand. Also, the regular update needs to reach the design phase also, to develop the newest sheaths against any possible breach of security.

VI. Conclusion & Future Work

The omnipresence of web applications and e-learning tools makes it vital to ensure that the tools are secure, correct and efficient. It is the opinion of the authors that no final "score" can be assigned to each platform tested, nor we can structure a table with the platforms tested in an ascending order based on such criteria since there is no widely accepted tool or technique available today that assigns a specific vulnerability to a "severity" scoring number. If such a tool existed, we could sum up the scores of each of the vulnerabilities found for a specific platform and come up with a final score for each one. This would lead to a taxonomy that could be shorted.

The first impression from Fig. 3 is that some platforms are more "weak" than the rest since they seem to have more recognized vulnerabilities than the other applications. This is not always true: even one of the vulnerabilities sometimes is more than enough for the intruder to exploit and gain even full control of the application exploited. In some other cases, an application can have several vulnerabilities but none of them would lead to a serious compromise because of other security measures that it incorporates. The paper focuses on software "families" for security. Such work can help both

academia and the industry to understand the importance of the security aspect of software and help researchers and programmers to improve their products.

Another future research direction could be the systematic evaluation and taxonomy of vulnerability scanners according to detailed standards and rules. This will help all stakeholders to talk to a more or less "common language", obviously in accordance with security organizations such as OWASP, etc.

In (Njenga & Fourie, 2010) it is clearly stated that e-learning greatly improved the academic performance and successful completion of courses among academically stronger students. It would be very interesting for a dedicated researcher to investigate the relation between the above argument and the security problems of the various e-learning platforms.

REFERENCES

[1] Acunetix. (2014). Website Security with Acunetix Web Vulnerability Scanner. Retrieved October 7, 2014,http://www.acunetix.com/

[2] Attwell, G. (2006). Evaluating E-learning: A Guide to the Evaluation of E-learning. Evaluate Europe Handbook Series, 2, 1610–0875.

[3] Barth, A., Jackson, C., & Mitchell, J. C. (2008). Robust defenses for cross-site request forgery. In Proceedings of the 15th ACM conference on Computer and communications security (pp. 75–88). ACM. http://dl.acm.org/citation.cfm?id=1455782

[4] Bau, J., Bursztein, E., Gupta, D., & Mitchell, J. (2010). State of the art: Automated black-box web application vulnerability testing. In Security and Privacy (SP), 2010 IEEE Symposium on (pp. 332–345). IEEE. http://ieeexplore.ieee.org/xpls/abs_all.jsp?arnumber=5504795

[5] Chen Y and He W 2013 Security risks and protection in online learning: A survey. The International Review of Research in Open and Distributed Learning. https://files.eric.ed.gov/fulltext/EJ1017552.pdf

[6] Doupé, A., Cova, M., & Vigna, G. (2010). Why Johnny can't pentest: An analysis of black-box web vulnerability scanners. In Detection of Intrusions and Malware, and Vulnerability Assessment (pp. 111–131). Springer. http://link.springer.com/chapter/10.1007/978-3-642-14215-4_7

[7] El-Khatib, K., Korba, L., Xu, Y., & Yee, G. (2003). Privacy and security in e-learning. International Journal of Distance Education Technologies (IJDET), 1(4), 1–19.

[8] Ertaul, L., & Martirosyan, Y. (2013). Implementation of a Web Application for Evaluation of Web Application Security Scanners. http://world-comp.org/p2012/SAM9718.pdf

[9] Fong, E., & Okun, V. (2007). Web application scanners: definitions and functions. In System Sciences, 2007. HICSS 2007. 40th Annual Hawaii International Conference on (p. 280b–280b). IEEE. http://ieeexplore.ieee.org/xpls/abs_all.jsp?arnumber=4076950

[10] Fonseca, J., Vieira, M., & Madeira, H. (2007). Testing and comparing Web vulnerability scanning tools for SQL injection and XSS attacks. In Dependable Computing, 2007. PRDC 2007. 13th Pacific Rim International Symposium on (pp. 365–372). IEEE. Retrieved from http://ieeexplore.ieee.org/xpls/abs_all.jsp?arnumber=4459684

[11] Gluck, Y., Harris, N., & PRADO, A. Á. (2013). BREACH: REVIVING THE CRIME ATTACK. Online at Http://breachattack. com/resources/BREACH% 20-% 20SSL,% 20gone% 20in% 2030% 20seconds. Pdf. Retrieved from http://css.csail.mit.edu/6.858/2013/readings/breach.pdf

[12] Government of Canada. (2000). Personal Information Protection and Electronic Documents Act. Justice Dpt. Retrieved from http://laws-lois.justice.gc.ca/eng/acts/P-8.6/page-16.html#h-25

[13] Halfond, W. G., Viegas, J., & Orso, A. (2006). A classification of SQL-injection attacks and countermeasures. In Proceedings of the IEEE International Symposium on Secure Software Engineering, Arlington, VA, USA (pp. 13–15). Retrieved from http://www.cc.gatech.edu/fac/Alex.Orso/papers/halfond.viegas.orso.ISSSE06.pdf

[14] Huang, E. Y., Lin, S. W., & Huang, T. K. (2012). What type of learning style leads to online participation in the mixed-mode e-learning environment? A study of software usage instruction. Computers & Education, 58(1), 338–349.

[15] Hussain, S., Wang, Z., & Sun, C. (2011). A comparative study of open-source learning management systems. In Open-Source Software for Scientific Computation (OSSC), 2011 International Workshop on (pp. 86–93). IEEE.

[16] ISO/IEC 13335-1. (2004). Information technology. Retrieved from http://www.iso.org/iso/catalogue_detail.htm?csnumber=39066

[17] Jeghal, A., Oughdir, L., & Tairi, H. (2014). Politic of security, privacy, and transparency in human learning systems. Education and Information Technologies, 1–10. http://doi.org/10.1007/s10639-014-9336-6

[18] Liaw, S.-S. (2008). Investigating students' perceived satisfaction, behavioral intention, and effectiveness of e-learning: A case study of the Blackboard system. Computers & Education, 51(2), 864–873.

[19] Luminita, D. C. (2011). Information security in E-learning Platforms. Procedia-Social and Behavioral Sciences, 15, 2689–2693.

[20] Martin, B. (2014, September 5). CWE - Common Weakness Scoring System (CWSS). https://cwe.mitre.org/cwss/cwss_v1.0.1.html

[21] May, M., & Sébastien, G. (2011). Privacy Concerns in E-learning: Is Using Tracking System a Thread. International Journal of Information and Education Technology, 1(1), 1–8.

[22] MITRE. (2011a). CWE - 2011 CWE/SANS Top 25 Most Dangerous Software Errors. http://cwe.mitre.org/top25/#Listing

[23] MITRE. (2011b). CWE - Common Weakness Scoring System (CWSS). https://cwe.mitre.org/cwss/

[24] Netsparker Community Edition. (2014). False Positive Free Web Application Security Scanner. www.netsparker.com/communityedition/

[25] Njenga, J. K., & Fourie, L. C. H. (2010). The myths about e-learning in higher education. British Journal of Educational Technology, 41(2), 199–212.

[26] N-Stalker. (2014). The Web Security Specialists. Retrieved July 28, 2014, from http://www.nstalker.com/

[27] OWASP. (2013). Top 10 2013. https://www.owasp.org/index.php/Top_10_2013-Top_10

[28] PCI. (2014). Approved Scanning Vendors. https://www.pcisecuritystandards.org/approved_companies_providers/approved_scanning_vendors.php

[29] Shay, C. (2014). Security Tools Benchmarking. http://sectooladdict.blogspot.ca/

[30] Violettas, G. E., Theodorou, T. L., & Stephanides, G. C. (2013). E-Learning Software Security: Tested for Security Vulnerabilities & Issues. In e-Learning" Best Practices in Management, Design and Development of e-Courses: Standards of Excellence and Creativity", 2013 Fourth International Conference on (pp. 233–240). IEEE. http://ieeexplore.ieee.org/xpls/abs_all.jsp?arnumber=6745547

TABLE IV. VULNERABILITIES FOUND BY NETSPARKER IN THE EXAMINED E-LEARNING PLATFORM

E-Learning Platform	Dokeos	ILIAS	Atutor	Sakai	Moodle	Claroline	Docebo	eFront	OLAT	Chamilo	Canvas	LAMS	LON-CAPA	CommSy	Fedena
VULNERABILITY															
IMPORTANT - CRITICAL															
SQL Injection														√	
Cross-site Scripting	√	√	√			√						√	√	√	√
Password Transmitted over HTTP	√	√			√	√	√	√	√			√			
Basic Authorization over HTTP	√														
SVN Detected		√													
Cookie Not Marked as Secure			√							√				√	
MEDIUM															
Possible Source Code Disclosure	√		√			√					√				
HTTP Header Injection				√											
GIT Detected					√										
Open Policy Crossdomain.xml Detected							√								
Password Transmitted over Query String												√	√		
LOW IMPORTANT															
Auto Complete Enabled	√	√	√	√	√	√	√	√	√	√	√	√	√		√
Cookie Not Marked as HttpOnly	√	√	√		√	√	√	√	√	√		√		√	
Information Disclosure (Microsoft Office)	√														
Cross-site Request Forgery in Login Form Detected	√	√	√	√	√	√		√	√	√		√			√
Cross-site Request Forgery Detected	√	√	√		√	√		√	√	√	√	√	√	√	√
Internet IP Address Disclosure	√														
Backup File Disclosure	√														
Cookie Values Used in Anti-CSRF							√								
Programming Error Message						√									
TRACE/TRCK Method Detected						√		√					√		
Insecure Frame (external)										√					
Database Error Message Disclosure														√	
INFORMATIONAL															
E-mail address disclosure	√	√	√	√		√				√		√	√	√	√
Internal Path Disclosure	√		√				√			√	√	√	√	√	

TABLE V. VULNERABILITIES FOUND BY ACUNETIX IN THE EXAMINED E-LEARNING PLATFORMS

E-Learning Platform	Dokeos	ILIAS	Atutor	Sakai	Moodle	Claroline	Docebo	eFront	OLAT	Chamilo	Canvas	LAMS	LON-CAPA	CommSy	Fedena
VULNERABILITY/SEVERITY															
IMPORTANT - CRITICAL															
SQL Injection														√	
Blind SQL Injection														√	
Cross site scripting	√	√				√	√						√	√	√
JQuery cross site scripting	√												√	√	
Directory traversal				√										√	
BREACH attack											√			√	
FCKeditor spellchecker.php XSS vulnerability														√	
Vulnerable JavaScript library[1]	√						√						√		√
Git repository found					√										
CRIME SSL/TLS attack				√											
CRLF injection/ HTTP response splitting				√											
File inclusion				√											
MEDIUM															
Application error message				√		√	√				√			√	
Error message on page	√										√				
Source code disclosure											√				
HTML form without CSRF protection	√	√	√	√		√			√	√					
Insecure transition from HTTP to HTTPS in form post	√		√												
User credentials are sent in clear text	√	√	√			√		√							√
Password field submitted using GET method	√														
Directory listing	√														
Basic authentication over HTTP	√														
Insecure crossdomain.xml file							√								
LOW IMPORTANT															
Clickjacking: X-Frame Options header missing	√			√	√					√			√	√	√
Insecure transition from HTTPS to HTTP in form post														√	
Possible sensitive directories	√				√			√					√	√	
Possible sensitive files					√			√							
Session Cookie without HTTP Only flag set	√	√	√		√	√	√	√	√	√		√		√	
Session Cookie without Secure flag set	√	√	√		√	√	√	√	√	√		√		√	√
Insecure Flash embed parameter			√								√				
Documentation file					√										
Login page password-guessing attack	√		√	√		√	√								√
Sensitive page could be cached	√		√					√							
Session token in URL	√														
Ruby on Rails CookieStore session cookie persistence															√
INFORMATIONAL															
Error page web server version disclosure	√	√	√				√			√			√	√	
Password type input with auto-complete enabled	√	√	√	√		√	√	√	√	√	√		√	√	√
Broken links	√		√			√				√	√		√	√	√
Possible Server Path Disclosure	√							√						√	
Email address found					√						√		√		
Possible username or password disclosure	√		√				√				√				
GHDB: Files uploaded through FTP	√									√					
GHDB: Apache Tomcat Error message				√											
Content type is not specified	√		√												
GHDB: 500 Internal Server Error	√														
GHDB: Internal Server Error	√														
GHDB: Possible sensitive directory	√														
GHDB: Possible upload script	√														
GHDB: Mp3 file	√														
Possible internal IP address disclosure	√														

TABLE VI. SEVERE VULNERABILITIES FOUND BY BOTH SCANNERS

(Where possible, vulnerabilities were included under one "super" category)

E-Learning Platform	Dokeos	ILIAS	Atutor	Sakai	Moodle	Claroline	Docebo	eFront	OLAT	Chamilo	Canvas	LAMS	LON-CAPA	CommSy	Fedena
SQL Injection of any kind														√	
Cross Site Scripting of any kind	√	√	√			√	√					√	√	√	√
Directory traversal				√										√	
BREACH attack											√			√	
Vulnerable JavaScript library	√						√						√		√
Git repository or SVN detected		√	√		√										
CRIME SSL/TLS attack				√											
CRLF injection/ HTTP response splitting				√											
File inclusion				√											
HTTP parameter pollution		√													
Authentication of any kind over HTTP	√	√			√	√	√	√	√			√			
Cookie Not Marked as Secure			√							√				√	