SECURITY IN E-LEARNING: INTEGRATED USER-CENTRIC APPROACH

MARJAN MILOŠEVIĆ
University of Kragujevac, Faculty of Technical Sciences Čačak, marjan.milosevic@ftn.kg.ac.rs

DANIJELA MILOŠEVIĆ
University of Kragujevac, Faculty of Technical Sciences Čačak, danijela.milosevic@ftn.kg.ac.rs

Abstract: *The paper presents augmented LTSA e-learning architecture model, introducing a security layer. In order to foster user position in terms of security, the standard architecture is upgraded with a security agent that provides constant monitoring of user behavior and feedback in terms of important security events. Agent's primary role is advisable: it is supposed to educate user and moderate his behavior. The model is presented both with implementation concept in Moodle LMS.*

Keywords: *e-learning, security, Moodle*

1. INTRODUCTION

Security is an ever-present issue in any e-business, including e-learning. Still, most effort is put in other areas, such as e-banking, often neglecting other e-domains[1].

However, E-learning market is constantly increasing. As reported by Docebo [2], the Asian market is showing the highest annual growth rate, at 17.3%; followed by Eastern Europe, Africa, and Latin America at 16.9%, 15.2%, and 14.6%, respectively. Among the main stakeholders, there are startups, large companies and universities. E-learners are everywhere: among tech savvy people, but also among not so IT-profound users.

Security is not mainly about technology. Further, it is reported that human factor is among topmost influential features in security [3]. Human aspect is therefore built into every modern security architecture. Among "ten deadly sins of information technologies" human takes several seats [4].

However, certain efforts are put into research of security in e-learning, but pointing mostly to the technology issues. There are some exceptions, such as [5].

E-learning standards very briefly introduce security concepts, if at all [6]. Standards mainly tackle issues of content interoperability and stays in education domain.

Idea of this paper is to bring together importance of taking care of security in e-learning and the "human" fold of the security concept. It is supposed to be done by extending the LTSA architecture with a security layer taking care of user actions and guiding and advising him through usage of the learning system. In order to achieve the goal, the user profile should be equipped with required information and it is recommended to be kept in a portable format.

Further on, we will discuss implementation issues on the Moodle LMS example.

2. SECURITY RISKS IN E-LEARNING

Having a complex system involving many users, ICT infrastructure and LMS, we may identify various risks in e-learning. Some are general and not particularly related to the e-learning systems, but encompassing wider Internet security issues. On the other hand, there are specific risks. Weippl gave a systematized preview of those risks[7]. A list also might be found at Zuev, who stated the following threats categories [8]:

- unauthorized access to digital content including physical access to servers,
- loss of integrity, and inadequacy of educational resources
- violation of assessment procedures,
- violation of the normal functioning of the e-University's departments and services,
- violation of the law (mostly copyrights and other rights)

Many risks are user related: user may cause a risk, user may be jeopardized by risk or he may report a possible risk. The user behaviour might be regulated by the e-learning policy that every user should read, agree and comply with. However, requiring user to click "I agree" surely will not guarantee he would comply with the policy. It will not even assure he read it at all. Research showed a substantial gap between users declared security behaviour (their intentions) and the real expressed behaviour [9]. Therefore there should be a mechanism of continuous monitoring of user behaviour in order to prevent unwanted situations and to increase user security awareness. That means bridging the gap between user's intention and his action, with advisory mechanisms involved.

3. SECURITY MEETS E-LEARNING STANDARDS

E-Learning has been standardized by numerous organizations, but, as we previously stated, they do not substantially deal with security.

Some standards are pointed towards learning process and portability of learning objects solely, while other also treat the context of e-learning, including infrastructure [10]. A comprehensive preview of standards featuring e-learner security and privacy is rendered by Jerman-Blažič and Klobučar [6].

IEEE 1484 introduced LTSA (Learning Technology System Architecture), a very generic, high-level e-learning architecture model [11]. The architecture provides a capability of modelling a broad range of learning scenarios, even not being limited to e-learning. It brings the possibility to adapt it or upgrade it easily, so it represent the desired scenario. It may be individual learning, group online learning or even traditional classroom scenario.

LTSA is constituted of entities and processes. Example of entity is Learner. It is abstraction of human learner and also may represent group of learners or learner in a specific role. Example of process is Delivery. It transforms the located Learning Content into a Multimedia presentation.

Upgrading the LTSA

The layer of upgrade has role of getting relevant data from the basic architecture setup, updating the user model according to the observed user behaviour, modifying certain existing security controls and improve user security awareness (Image 1). The main process in charge of dealing with information presented to the user is still Delivery. User may be equalized with "Learner entity" in this model. We do not want to neglect that other classes of users exists, such as teachers and administrators, but this model focuses on the major population - learners. Nevertheless, model may be applyed to general users too.

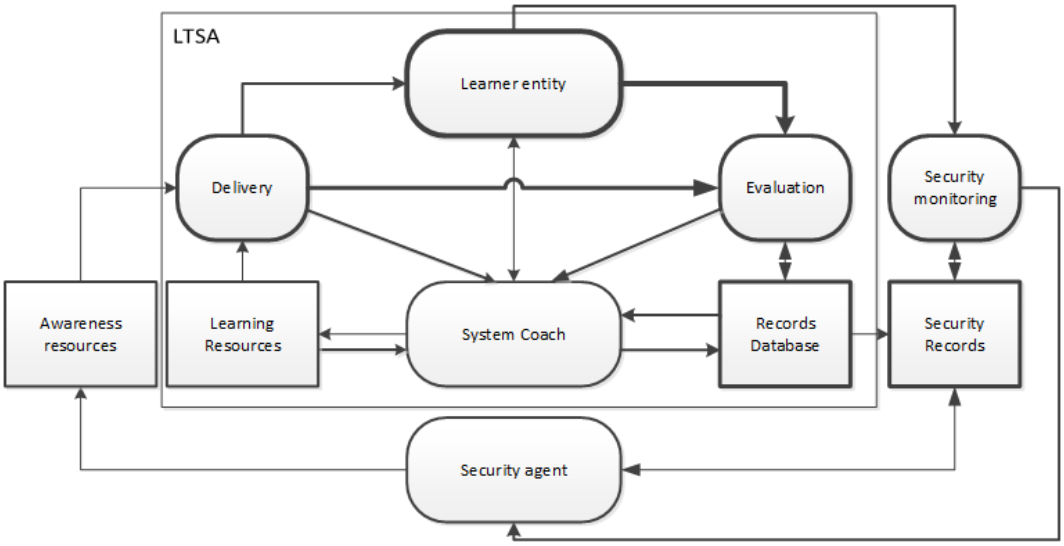


Image 1: Upgraded LTSA

The Delivery embraces all actions taken in order to adapt the learning system behavior in front of the user. That means providing additional information regarding access rights, preserving privacy or emphasizing parts of user interface.

The central part of the upgrade is the Security Agent, which actually coordinates the whole process. It acts according to learner profile and log analysis: it advises learner and direct him to certain contents. Such contents may be incorporated in the learning content (i.e. in course), or it can be a distinct resource that user should encounter.

As we mentioned, the learner is what the actual human learner represents. Usually we have various information about user in institutionalized e-learning forms (i.e. at universities' learning systems), such as phone numbers, address, interests and social network info. Beside that data, there are records about academic achievments (test scores, badges, assignments) and system data such as access times or errors. In order to maintain additional data, related to security issues, we introduced additional fields, which should improve the original profile. These fields are meant to be rather simple and to keep essential data about user "awareness achievment" or "personal status" in security context. The basic user model thus has three additional elements: login administration, privacy control and general security capacity. Login administration is supposed to describe user profficiency in this area, according to possible login issues. For example, if a user shows tendency of login failure, often procede with password recovery or forget to logout and so on, the "login administration" will keep the qualifications as levels of awareness in the particular field. Then the agent may act accordingly, providing user with hints for creating strong, yet memorable passwords.

Privacy control keeps a user preference of what personal data is available to which users, in specific context. It is meant to allow user to control the field by themselves, but also to let the system accommodate the data visibility automatically (default option).

Security capacity is a category describing a general security awareness and practice of a user. It is implementation-dependent and might include things like message spamming, bad url guessing, mass downloading, inappropriate content in profile, uploading unsafe files and so on. It is important to document the specific instance (for certain platform implementation) so it could be implemented in a functional way.

The user profile is formed based on the observed behavior, reported in [12].

4. IMPLEMENTATION ISSUES

We will discuss implementation in Moodle [13] system. Moodle is chosen because it is open source and it is already used at the authors' faculty.

Moodle is a modular platform, available in over 170 languages and it has a large community. Its modularity means that additional funcionalities may be implemented in form of plugins - actually majority of its built-in funcionalities is also realized as plugins.

In order to implement the upgraded LTSA model using Moodle, one should analyze Moodle's architecture and map required resources and processess to the Moodle existing elements and define which elements are supposed to be added.

Moodle Architecture

On a higher level, Moodle architecture follows a traditional 3-tier web-application design.

Roughly speaking Moodle is consisted of core and modules (image 2). Core got all the basic libraries and defines API that all other parts use. Modules may be defined in several categories, such as: block, activities, themes, repositories and so on. Following the defined way of how to create, install and maintain modules, brings them in a fully decoupled fashion. On next level, it is administrator's job to install and enable the module and after that there is up to either administrator or teacher to use certain module, depending of its type and purpose. For instance, if a Youtube repository is enabled, then teacher may browse and embed youyube videos directly from Moodle.

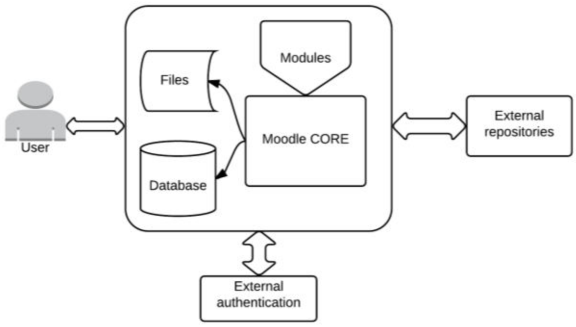


Image 2: Moodle architecture

User's interaction with the system is through a web-browser, while choice of particular database and web-server is implementation-dependant.

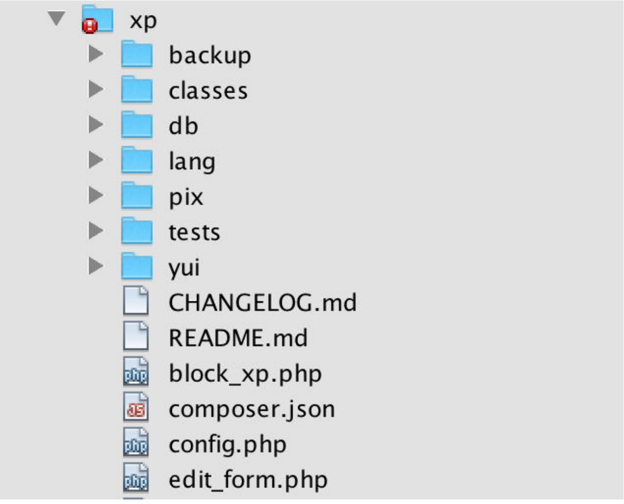
Closely to the learning platform, we may differentiate various types of modules: course types, authentication types, blocks, activities, reports and so on. It is possible to add many features by fully modular approach. Enabling proper add-in functionality is the recommended way of enriching Moodle with new possibilities. In that way the update process will not overrun any code. Also, pluggable architecture enables independent modules update. However, one may not always do so: if there is a need for modifying certain features incorporated in core, the very core modification is required.

Plugin architecture

It is very important to follow the recommendation Moodle team gave in order to design plugin in fully modular fashion. Moodle got high level of abstraction, which means that there is rarely if ever needed to directly use database or set web-page elements. There is a powerful API that is supposed to be used. For example, in order to make new tables in database, the appropriate way is to define XML table model (using the Moodle embedded tool), put the code into install file and trigger the installation.

As said, plugins (modules) may be realized as various Moodle elements and most popular are activities and blocks. Activities are stored mostly in the mod directory and represent different learning content elements, such as quiz or assignment. Blocks contain visual, not so dynamic contents, positioned aside the main site content.

Image 3 shows a typical plugin structure: there is mandatory db folder with install file and message configuration and regular files with plugin libraries and core functionalities.



- xp
  - backup
  - classes
  - db
  - lang
  - pix
  - tests
  - yui
  - CHANGELOG.md
  - README.md
  - block_xp.php
  - composer.json
  - config.php
  - edit_form.php

Image 3: Plugin structure

Also, there are language folder and eventually classes folder, where, beside other things, events are defined.

Security Layer on Moodle: Analysis

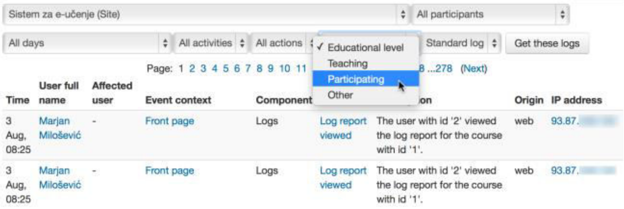
In order to implement the whole layer and especially the agent, it is needed to analyse how the model maps to the existing features Moodle got and which features lack and need to be added.

User profile

In order to maintain security data, we need to enrich the user profile for three fields. As mentioned, since it is recommended not to deal with database directly, these fields can be added through the plugin installation. Also, in order to avoid altering the original user table (mdl_user by default), an additional table should be added that match one-to-one to the basic user table. These fields would be set to default values and afterward, the agent would update their values according to its logic.

Getting data about user activity

In order to monitor the user and act accordingly to his behaviour, it is necessary to get data from logs and to trigger certain important events. Moodle logs have been redesigned in version 2.7 and now provide a flexible way of getting more log details and define new types of events (image 4).



Time	User full name	Affected user	Event context	Component		on	Origin	IP address
3 Aug, 08:25	Marjan Milošević	-	Front page	Logs	Log report viewed	The user with id '2' viewed the log report for the course with id '1'.	web	93.87.[redacted]
3 Aug, 08:25	Marjan Milošević	-	Front page	Logs	Log report viewed	The user with id '2' viewed the log report for the course with id '1'.	web	93.87.[redacted]

Image 4: Moodle logs

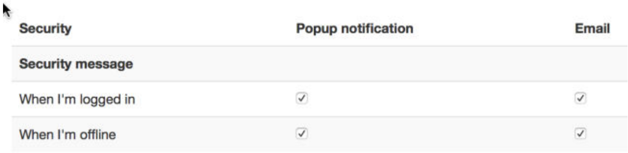
Logs, in this original form, can be used for gathering data about failed logins, password reset, personal message sending, often IP address changing and other events that could indicate behaviour anomaly. Also, logs can be used in order to write details of security agent events, such as profile update, message sending and so on.

Moodle currently supports three event categories: Teaching, Participating and Other. Every event that would be added on behalf of security agent should be put into category "Other" in order to align with existing Moodle event taxonomy. For example, if user gets a message directing him to some security awareness material, an event should be triggered and appropriate log line should be written to log.

Communicating with the user

Important part of agent functioning is directing the user to various ways he can improve his security. It can be done by block-module, which content would adapt to the current user security profile. For instance, if the user has recently registered, he got default profile values and is supposed to get pieces of security policy throughout the system usage. Then the block is set on his courses that shows pieces of policy every time the user access a course. Surely, the block is not supposed to be too large, or to take dominant place on the course, but should of stand as a note a user would occasionally read. Beside that, block may present different contents that suit the current user profile. Also, it might point to a test a user should take in order of possible profile upgrading.

Moodle got two basic ways of contacting certain user: by e-mail and by personal messages. Module itself can send messages: it is required that message is defined in the special module file and that certain functions are called in order to send it. Messaging preferences can be set on user-level (Image 5).



Security	Popup notification	Email
Security message		
When I'm logged in	☑	☑
When I'm offline	☑	☑

Image 5: User messaging preferences

Using the mail and messaging, system can communicate with user, sending him valuable information such as security policy, privacy strategies, password generating and directing him to the awareness assessment page or other external resources.

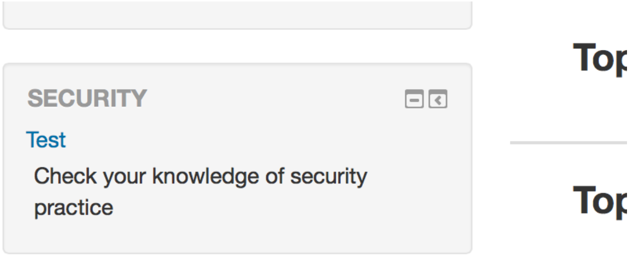
Site adaptation

There are several things that can be adapted according to user. For instance, if user tends not to logoff, leaving intruder a chance to continue using the site on logged account, then the logoff link may be emphasized, with proper message aside telling why is it important to end session, or not remember credentials on public computer.

The user privacy settings also may be adapted, in order to control their visibility.

Agent architecture

Since there are various elements that are up to be monitored, there is hardly possible to conduct the layer realization without altering the core code. For example, if we want to catch event of user uploading an infected file, we need to add appropriate code to a particular core file (manager.php). Therefore, the agent should be built in a hybrid fashion, as classic module (block) and as core hack. Block would take care of messaging the user, presenting the content and directing him to external resources, while core hack would deal with virus-detection, main interface changing and other issues, not manageable on module-level.



Check your knowledge of security practice

Image 6: Block in action

On Image 6 an example block-content is shown. It's a link to the Security awareness test. User is provided with link that is option to follow, leading to a Moodle quiz that assess his knowledge of security basics, parts important for e-learning. The result is taken into count while upgrading the user profile. If the awareness level is high, the security capacity will rise, otherwise it will go down. The test can be taken again after while, so user can get better.

Changing the core needs to be followed by extensive documentation since severe upgrading issues may arise. For instance, if virus-handling functions get moved from the core level to plugin level in next version, a proper change needs to be made in order to keep the agent working after upgrade.

5. CONCLUSIONS AND FUTURE WORK

Human is a key figure in complex process called security and every kind of information system security architecture incorporates human as a building block. LTSA architecture can be easily upgraded in order to add functionality such as user security control. That role is forwarded to an agent, main part of the added layer. The layer fits in LTSA since its primary task is to advise and educate learner and the very learning environment is just place for such thing.

Implementation in Moodle brings certain obstacles, since the agent is complex and not all features can be incorporated in modular fashion. Therefore a core hack is required.

Future work means full implementation and putting the agent in test phase on a live platform. Runtime issues should be documented. User response to the agent is supposed to be monitored and we will conduct an evaluation in order to check how the agent works. Additional reported issues will be examined and incorporated in next development iteration

LITERATURE

[1] Frank Graf, *Providing Security for E-Learning,* Computers and Graphics (26), pp 355-365, Elsevier, 2002

[2] *E-Learning market infographic* https://www.docebo.com/2014/04/18/infographic-elearning-market-2014-2016/?SOCIAL-LINKEDIN (accessed in july 2015)

[3] Basie von Solms and Rossouw von Solms, The 10 deadly sins of information security management, *Computers and Security* (23), pp 371-376. Elsevier, 2004

[4] Jan Eloff and M.M.Ellof, Information Security Architecture, *Computer Fraud and Security*, pp 10-15, IEEE, 2005

[5] Z. F. Zamzuri, M. Manaf, Y. Yunus, and A. Ahmad, “Student Perception on Security Requirement of e-Learning Services,” *Procedia - Soc. Behav. Sci.*, vol. 90, no. InCULT 2012, pp. 923–930, 2013.

[6] B. Jerman-Blažič and T. Klobučar, “Privacy provision in e-learning standardized systems: status and improvements,” *Comput. Stand. Interfaces*, vol. 27, no. 6, pp. 561–578, Jun. 2005.

[7] E. R. Weippl, *Security in e-learning*. New York, NY: Springer, 2005.

[8] V. I. Zuev, “E-Learning Security Models,” *Manag. Inf. Syst.*, vol. 7, no. 2, pp. 24–28, 2012.

[9] C. Jensen, C. Potts, and C. Jensen, “Privacy practices of Internet users: Self-reports versus observed behavior,” *Int. J. Hum. Comput. Stud.*, vol. 63, no. 1–2, pp. 203–227, Jul. 2005.

[10] L. E. Anido-Rifón, M. J. Fernández-Iglesias, M. Caeiro-Rodríguez, J. M. Santos-Gago, M. Llamas-Nistal, L. Álvarez Sabucedo, and R. Míguez Pérez, “Standardization in computer-based education,” *Comput. Stand. Interfaces*, vol. 36, no. 3, pp. 604–625, Mar. 2014.

[11] V. Devedzic, J. Jovanovic, and D. Gasevic, “The Pragmatics of Current E-Learning Standards,” *IEEE Internet Comput.*, no. June, pp. 19–27, 2007.

[12] M. Milosevic, R. Krneta, and D. Milosevic, “Security and Privacy in On-Line Learning: Case Study from Serbia,” *Metal. Int.*, vol. 18, no. 3, pp. 85–88.

[13] M. Dougiamas, “Moodle.” 2011. URL: moodle.org (accessed in july 2015)