Security audit and mechanism of protecting e-Learning system at the Faculty of Transport and Traffic Sciences

Peraković, D., Remenar, V.
Fakultet prometnih znanosti, Vukelićeva 4, 10000 Zagreb
dragan.perakovic@fpz.hr, vladimir.remenar@fpz.hr

Abstract. *LMS (Learning Management System) of the Faculty of Transport and Traffic Sciences, called e-Student, was experimentally introduced at the end of 2004 for one subject at the Faculty and was used to carry out a part of the teaching and practical work for about a hundred students. Today the system is used by more than 4800 students. In the period from 2004 to the beginning of March 2007, the system was used more than 145,000 times. With the fact that e-Student system is a publicly accessible web application has given rise to questions regarding the security of the users interface and the database safety. Although, from the very beginning the system was planned and designed so as to provide security against then known methods of attacks, there are almost daily new failures in the operating systems and database management systems and the methods of attacks and usage of the web application drawbacks. Consequently, the system has to be regularly tested and adequately protected.*

1. Introduction

Web applications are very often the vulnerable part of the information system. The very nature of web applications, expressed through their openness, accessibility and wide distribution, makes them an extremely interesting target to malicious users. The security problem of web applications is increasing because of the increasing complexity of web applications and an increasing number of development technologies and programming languages for the development of applications.

Attacks on small and rarely used web applications are not interesting since the hackers cannot boast of their "success" nor can they have direct material benefits. The growth of the web application means at the same time the growth of its popularity which makes it the potential target of malicious users.

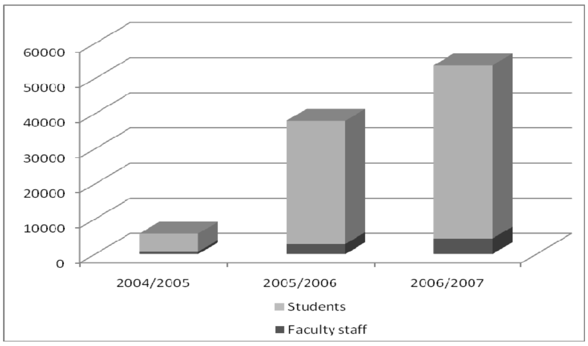
Very often the security of web applications is neglected, either because of insufficient skills of the programmer, short time deadlines for the development and starting to operate, and insufficient finances for analyzing and protecting the web application. According to the Laboratory for systems and signals at the Faculty of Electrical Engineering and Computing, as much as 90% of web applications have one or more of the many security errors, whereas more than 75% of system attacks occur at the application level.

The work presents the most commonly used methods of testing and protecting web applications used by designers and administrators of the e-Student system in order to protect adequately and timely the LMS system of the Faculty of Transport and Traffic Sciences, against malicious users and those who may violate the stable and safe functioning of the system through their ignorance and carelessness.

2. Analysis of FPZ LMS system application

LMS of the Faculty of Transport and Traffic Sciences was introduced experimentally in the teaching process for one syllabus at the end of 2004. At that time the system was used by some hundred students and this way of teaching and practical work was a supplement of the classical way of the teaching process. Today, the system is used by more than 4800 students who accessed the system in the last three and a half years more than 145,000 times.

With every new academic year, there is an increase in the number of accesses by students and the teaching staff members, which can be seen in Graphs 1 and 2. It is obvious that the number of accesses of students in the fall semester of the academic year 2004/2005 amounted to 5166 accesses. Already in the fall semester of the academic year 2005/2006 the number of student accesses to the system increased to 35,098 accesses, and in the fall semester 2006/2007 this number amounted to 49,296 accesses. The same trend of growth was also in the number accesses by the teaching staff, so that in the fall semester of 2004/2005 six hundred and sixty-five accesses were realized whereas in the academic year 2006/2007 in the fall semester a number of 4416 accesses to the LMS system of the Faculty of Transport and Traffic Sciences, was realized.



Graph 1 – Number of accesses to the e-Learning system of the Faculty of Transport and Traffic Sciences

3. Security auditing methods

3.1 Auditing techniques

Four techniques of application security auditing may be carried out: manual analysis, static analysis (which is divided into the analysis of source code and the analysis of compiled files), dynamic analysis and fuzzing methods.

Manual analysis is carried out by using standard and quite available tools, e.g. using the Internet Explorer. This method is used to manually generate the input values that are forwarded to the tested application. During the manual analysis the output data from the application and the very behavior of the application are monitored. The manual analysis method is very inefficient and it is almost impossible to discover the more complex and more hidden vulnerabilities of the web applications and as such serves only as the preparation for more complex and advanced tests.

Static analysis can be used in cases when the source code of the web application or compiled web application files can be accessed. Very often the access to the source code is not possible so that the only possible analysis is the analysis of the compiled files. By analyzing the compiled files it is possible to find the forgotten commentaries made by web application authors, which may contain various data, from useless instructions to data that are used to connect to the server or SQL database. If source code of the web application can be accessed, it is much easier to find the drawbacks or security flaws within the application. Static analysis of the source code of the application makes it possible to discover the hidden and very complex vulnerabilities; however, it is very time-demanding and requires knowledge of the programming language which was used to develop the application.

The dynamic analysis is carried out during the execution of the application. During the execution of the application the input and output data are monitored, trying to identify the uncommon conditions of the application and the application error handling. Bringing the application into the condition for which it was not meant identifies the drawbacks in the development of the application whereas insufficient or inappropriate error handling can identify the otherwise inaccessible data.

The fuzzing method of application auditing is based on the generation of a huge volume of random data of different sizes and formats and forwarding of these data to different application inputs. In order to implement this method, specialized tools or one's own developed applications are used. As with dynamic analysis, the status of the application is monitored as well as the error handling possibilities of the application. Since this method generates a large volume of data and forwards this volume to the application, it is often impossible to perform this method since the testing only can be classified s *brute-force* attack on the system.

3.2 Penetration auditing

Penetration auditing is used to evaluate in detail the security status of the entire system, from the operating system on which the web application is executed, over the components of the operating system itself and various additional applications installed in the operating system (such as FTP server, etc.), database auditing and auditing of the web application itself. Each of these components is a potential weak link and if one wants to completely revise the security status it is necessary to perform complete test of all the components. The fault in one of the components may lead to instability or the complete stoppage of the web application. During penetration auditing all the techniques and tools are used that a users might use if they wanted to disrupt a stable and safe running of the application. The result of testing is a detailed insight into the possible flaws and/or security faults in the e-Learning system of the Faculty of Transport and Traffic Sciences. Based on these results it is possible to solve the flaws and errors within the system.

3.3 Web application auditing

In order to adequately protect the integral information system of FPZ, with complete penetration auditing, it is necessary to perform specialized and adjusted testing. Since LMS system of the Faculty of Traffic and Transport Sciences is a web application it is necessary to perform penetration auditing specialized for web applications. The web application auditing includes audtiting to known vulnerabilities such as: SQL code injection, XSS vulnerability (Cross Site Scripting), manipulation of input parameters, error handling, listing of maps, discovery of shortcuts, etc. Web application auditing is used to evaluate the complete security status of the application in order to rapidly and safely eliminate possible drawbacks and errors.

3.4 Database auditing

The access to the data stored in databases must be strictly controlled since these are real-time data that describe the teaching process and eventually form the electronic folder on each student and his/her overall results during the study. Unauthorized access to data can cause serious consequences in assessment of the student engagement (the issue of ECTS credits assessment) and final success.

Since database servers are usually protected by firewall and thus inaccessible by direct access to the users from the Internet, thus protecting also the database of the LMS system of FPZ, these tests are carried out on the internal computer network. However, web application operation with database is also tested by means of tests of the right of access of the application to the very database.

4. Methodology of FPZ LMS system protection

4.1 Preliminary protection

4.1.1 Logical topology of FPZ information-communication network

The protection of information systems starts with detailed planning of the computer network and its protection. Inadequate planning of the network jeopardizes the work of all the computers and services within the network, including the operation of the web application executed within the respective network.

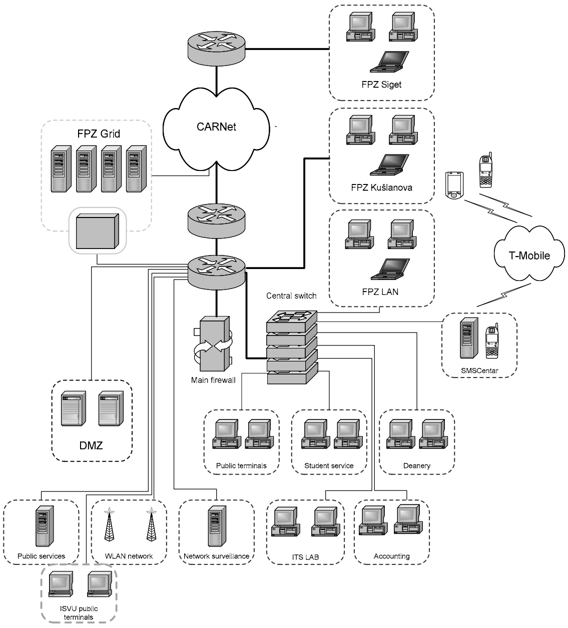


Figure 1 – Logical topology of FPZ information-communication network

In accordance to the security policy of the Faculty of Traffic and Transport Sciences and CARNet as well as the requirements for safe and reliable operation of the computer, servers and services in the Faculty network, logical topology of the Faculty network has been developed and established, as presented in Figure 1. The Faculty network protection is based on the logical separation of the computers into virtual local networks according to their purpose (student services, accountancy, public terminals, teaching staff, evidence, etc.) and the protection by means of firewall for Internet access. The distant locations of the Faculty are connected by virtual private networks and logically separated from the Faculty local network.

4.1.2 Checking files

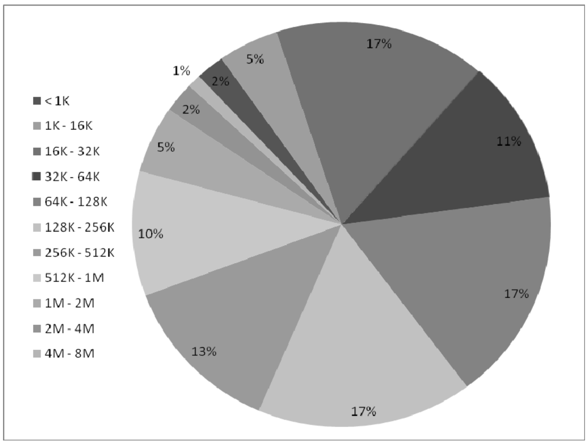
By means of the LMS system of the Faculty of Transport and Traffic Sciences, the students and the staff can publish various files necessary to give lectures, practical work, exercises, etc. Since files may cause damage to the system, all the files need to be thoroughly checked before storing them on the server.

The sent file is first checked regarding its format, i.e. no executive files are allowed (exe, com, dll, scr, etc.) and such files of inadequate format are automatically discarded.

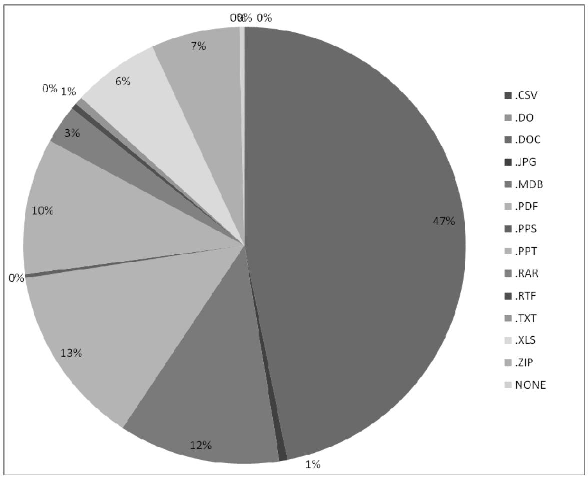
In order to prevent attacks on the server, which would interrupt and/or drastically slow down the operation of the server, the file size sent by the users needs to be limited. If the file size exceeds the maximal allowed file value, the system rejects such a file.

It is possible to hide a malicious application in the form of a virus which is why every file has to be checked for viruses. Although it is possible to compress the virus-infected file several times in order to hide the reference file, the antivirus application has the possibility of checking the compressed files of several compression degrees. Of course, for the antivirus application to be able to detect the hidden malicious applications, the antivirus program needs to be regularly updated by new definitions that contain virus signatures.

In defined time intervals, the files stored on the server are analyzed and the analyses are used to change the parameters of the permitted file size and format that can be stored by the users on the server. Graphs 1 and 2 show that the most common file format is the Word document in sizes from 64KB to 512KB. Based on the data presented in Graphs 2 and 3, the rules for permitted file sizes and formats can be defined. For instance, only 0.13% files are of size between 8 and 16MB and only 0.05% of format "csv" we can define a rule of prohibiting files of "csv" format and files bigger than 8MB.



Graph 2 – Percentage of file sizes stored on the server



Graph 3 – Percentage of types of files stored on the server

4.1.3 Data encryption

Encryption is modification of data using a defined key which is known only to the users who are allowed to read or modify the data. Individuals, who are not allowed to read or modify the data, either cannot or can with great difficulty read the encrypted data. Each data that represents a form of secrecy should be protected by encryption. In case a user who is not entitled to the data, even if s/he manages to obtain such a data, it is completely incomprehensive because of being encrypted.

The LMS system of the Faculty of Transport and Traffic Sciences uses the publicly available encryption method which is used to protect all the private and sensitive data. However, for greater security, the encryption method has been modified, thus obtaining a new version of encryption which is not publicly available.

4.2 Database protection

Almost all web applications fully rely on the stable and safe operation of the database management system. Unstable and unsafe database operation brings into question the operation of the entire system, and therefore, the database needs to be well designed and well protected.

The database is protected at several levels, and each database needs to be protected from the users which have the attributes of intention and the users whose ignorance and careless work can damage the database integrity.

High-quality protection of the database starts by locating the database behind the firewall in order to prevent direct access to the database from the Internet. With this procedure the database is protected against attempts of brute force method of finding the users name and password necessary to use the database, disabling the Denial of Service attack by wrong and over-dimensioned queries.

Correct adjustment of users accounts and remote access are very important items for good protection. Disabling remote access and limiting the access to local computers solve a large part of security problems, whereas correct adjustment of the database users rights prevents the change of data to which the users have no rights. Good setting of the users accounts includes the non-usage of users account “Administrator” and “sa” which have full privilege within the database, including execution of operating commands.

Database access should be ensured also through the web application itself in order to prevent SQLinject attacks that may destroy the data or start the operating system commands.

4.3 Protection within web applications

4.3.1 Authorizations levels

Within every real system there are also defined levels of management and decision-making. Consequently, LMS system of Faculty of Traffic and Transport Sciences has to be designed so as to recognize all the specific characteristics of the scientific and teaching organizations (Departments, Sections, assistants, etc.) that occur in the teaching process, performing practical work, monitoring of the work on seminar papers and projects, etc.

Compliance with the rules of authorization levels in information system management should prevent access to all the users to all parts of the system. Correctly authorized users regarding the system have to be limited to parts of the system they use so that they would not cause unstable system operation due to their ignorance or accidental procedures. This method also reduces the possibility of disturbing the stable and safe operation of the entire system if the malicious user who manages to obtain the authentication data of one of the system users.

Seven groups of users are defined within the LMS system of the Faculty of Traffic and Transport Sciences, i.e. seven levels of system users authorization adjusted to the Faculty operation processes.

4.3.2 Automatic logging off the system

Great threat to the security of every web application is leaving the system usage session open by the legitimate system users. Using the system on public terminals is rare but possible and if the system user inadvertently stays active in the system there is the possibility of using the system for malicious purposes.

Every system based as web application should have the option of the user's automatic logging off the system within a defined time period of idleness. By carefully studying the behavior of the users in the system it is easy to obtain the data on the user’s idle time interval after which the system logs off automatically thus disabling unauthorized activities within the system.

By studying the behavior of LMS system users of the Faculty, while using the system, the times of automated log-off of the users from the system have been defined. The times of automatic log-off have been defined differently according to the authorization levels.

4.3.3 Error management

Irregular, insufficient or poor error management in the system operation discovers to the potential malicious user very important data that should not be visible to the users.

Within errors generated by the system and shown unprocessed to the users it is possible to find the data on the database, documents otherwise invisible, etc. Such data are most often used for SQLinject attacks that may cause very difficult consequences for the data in the system and the system operation itself.

The most frequent reason for insufficient error management is the sharing of the same server which serves also for the system development and exploitation. The correct error display helps the system creators; however, it also helps the malicious users in their intentions.

LMS system of the Faculty of Traffic and Transport Sciences uses separate production and exploitation servers. The production server is used for testing and development and has been configured to display detailed data on errors, whereas the exploitation server has its own defined error pages which do not show the error nor its description.

4.4 Implemented LMS protection against most common forms of attacks

4.4.1 Brute force

Brute force attacks are very simple attacks during which the specialized or on one's own developed applications that generate a huge volume of random character string.

It sends such data to another application in hope of finding out the correct character string such as users names or passwords which are the usual target of the attack using brute force tools. A very frequent method is the finding out of the users name or password or in bigger systems a smaller number of more frequently used passwords is selected and the users name guessed.

There are several methods of defense against brute force tools, and the most widely spread is the monitoring of the number of irregular queries. After five incorrect requests of the IP address of the user, the date and time are entered into the database, and in case of a request the records are checked and in case the record exists, the request is automatically rejected.

4.4.2 SQLinject

SQLinject is a frequently used method of attack on poorly protected information systems in the form of web application. Insufficient protection against SQLinject attack can result in a large number of destroyed or stolen data. The attack itself is very simple and it is performed by trying to insert SQL code into the publicly accessible parts of the web application, URL or templates (e.g. templates for input of users name and password) which will execute a certain SQL command.

The commands can range from the very simple ones (e.g. registering into the system) to very dangerous and complex commands that could copy or delete a very large amount of data, all the way to commands that start the commands from the operating system itself in which the web application is run.

Very efficient and simple protection against SQLinject attack is the checking of the users input and filtering of the forbidden characters and key words of SQL commands that are used to execute the SQL queries. This includes checking whether the users query contains special characters such as: --, ', =, .., ; and similar, and checking whether there are key words typical to run SQL commands such as: OR, SELECT, DROP, INSERT, DELETE, UPDATE, XP_ etc. In case one of the conditions for the existence of these characters or words is met, the user is automatically rejected, the IP address, date and time and the form of attack recorded into the database, and the web application administrator is informed about all this.

4.4.3 Cross-site Scripting, XSS

XSS is a very frequent and much underestimated attack which allows theft of cookies, taking over users identity, session hijacking, or even complete change of the web application design. Very few web applications are resistant to XSS attacks so that even Google was the target in October 2004 when XSS by attacking completely changed the design of the page so that it displayed information that Google was to require subscription for its services.

One of the possible attacks is performed by inserting JavaScript code into the legitimate address of the applications, and by sending such modified address to the otherwise legitimate system user. By loading such address, if the application is not protected, it is possible for the malicious user to send data that may be used for unauthorized access to the web application. An example of an attack is to send the user the address in the following form:
http://www.nekaadresa.com?id=1<script>alert(„XSS!“)</script>. This attack will only warn the user of the fault in the web application displaying the message.

An efficient method of protection consists, as also in the case of the protection against SQLinject attack, to checking whether the users query contains forbidden characters such as: <, >, ;, / and similar, and automatic filtering of these characters.

4.4.4 Buffer overflow

A frequent error of web application programmers is having confidence in the system users which includes insufficient checking of data input by the user into the system, which allows very wide span of attacks on the system including the Buffer overflow attacks. Buffer overflow attacks occur when a part of application, i.e. one function cannot process the amount of data required of that function. If the input data are larger than the space planned for processing of these data, the data flow over into another part of the memory location. Thus memory locations planned for some other operation are copied by these data and more often than not this attack causes complete termination of the system operation and the entire server.

The most dangerous form of this attack is when the input data flow over into the memory which is then used to select the execution of the next instruction. A skilful malicious user will use this failure to execute the malicious code which can cause a lot of damage, steal data or access the server itself, etc.

The attack is executed by inputting a very large amount of data, even more than 100,000 characters in one attempt. The data are input into the publicly available forms, such as the forms for system access or registration for using the system. If there are no checks on the limits of input size and the data are blindly copied into the memory without being checked.

The basis of protection against this type of attack is the lack of confidence in the system users, especially with regard to web applications. Although limits to the record sizes on the client's part are easily set, it is almost as easy to modify these limits. It is therefore necessary to additionally check the entire input of data on the server's side, and if there is a volume of data larger than the allowed one, the excess of data should be simply rejected, i.e. ignored.

4.4.5 Denial of Service – DoS

One of the inevitable consequences of executing the web application code program is that it requires a certain time for execution. For each call of the function which is part of the web application, the application requires a certain number of cycles of the processor unit in order to execute only the function. This opens up the possibility for the Denial of Service attack. If it is possible in the sufficiently short time to send to the server sufficiently large number of requests for function execution it is possible to overburden the server with these requests and it is simply not able to execute any other operation. This attack is called the Denial of Service. The attack itself does not destroy or copy the data but denies the possibility of normal operation of the web application.

Protection from this attack is a very difficult and complicated task. Even by using the multi-server systems it is not possible to avoid this attack completely, but it can only be somewhat alleviated. Probably the best way of protection from this attack is to use special tools, such as IDS – Intrusion detection systems, for the detection of this type of attack. Such tools will detect possible attack through long-term connections or unusual volume of similar or same network traffic thus being able to activate the firewall in order to prevent further network traffic from one or more IP addresses. However, it is very difficult to distinguish the DoS attack from the legitimate network traffic. Therefore, in setting the protection against this type of attack a lot of attention should be paid in order to avoid the impossibility of accessing the system by the legitimate users.

4.4.6 42.zip file

Attack on the information system by 42.zip file is performed by sending to the server a specially designed file of 42kB, which is also the origin of its name. The file is created and compiled so that its decompression occupies 4PB of space thus either terminating or drastically slowing down the operation of the server. Since every file sent to the server passes the antivirus protection where antivirus application decompresses the files, insufficient handling of this file may result in unwanted consequences for the operation of the web application server.

The LMS system of the Faculty allows sending of compressed files and therefore the file passes the checks according to the file format. In order to protect adequately the LMS system from this type of attack on the web applications, the files of exactly 42kB in size cannot be sent to the server while the files are checked for viruses by the tools verified to the possibility of handling such files.

5. Conclusion

The fact that e-Student information system based on the modules of the own developed web applications emphasizes the security problems expressed through the simple accessibility and the huge number of users.

Reliable operation and the high level of data security is the priority and not a by-fact in the operation, maintenance and development of new LMS system modules.

Security testing against the existing attack methods are continuously performed in order to identify on time the most hidden errors, failures and possible faults in the system operation. This protects the modules of application for e-learning, the users and their data stored in the databases used by LMS.

It is very important that the system protection is not limited to the protection exclusively of LMS only, but the organization, program and software security mechanisms of protecting the entire information and communication system of the Faculty should be implemented and applied, including also other services that are applied, such as ISVU or accountancy system, as well as the specific applications of the Faculty staff.

In order to achieve safe and reliable integral information system of Faculty of Traffic and Transport Sciences, permanent education of the teaching and non-teaching staff of the Faculty about the security and the risks of electronic operation and application of the digital signature in the teaching process is of extreme importance. This includes absolute and unconditional compliance with the adopted security policy of applying the information and communication infrastructure of the Faculty of Traffic and Transport Sciences.

6. Literature

Whittaker, J.A., Andrews, M. (2006), *How to break web software,* Addison-Wesley Professional, 978-0321369444, Boston

Gold, H., Peraković, D. (2006), *Sigurnosna politika Fakulteta prometnih znanosti*, FPZ, Zagreb

Kavran, Z., Peraković, D., Remenar, V. (2006), *The impact of introducing e-learning system in the teaching process at the Faculty of traffic and transport sciences*, Proceedings of The 17th International DAAAM symposium, Vienna

Jušić S. (2004), *Web application security*, authorized presentation, Laboratorij za sustave i signale, Zagreb