Czech University of Life Sciences Prague

Faculty of Economics and Management

Department of Information Engineering

Bachelor thesis

Database security of web pages

Author: Chernookov Ilya

Supervisor: doc. Ing. Vojtěch Merunka, Ph.D.

CZECH UNIVERSITY OF LIFE SCIENCES PRAGUE

Faculty of Economics and Management

BACHELOR THESIS ASSIGNMENT

Ilya Chernookov

Informatics

Thesis title

Database security of web pages

Objectives of thesis

Protection of database of web-page from external intervention. Protection of web-page's content changing without administrator's knowledge.

Methodology

Using a simple web-page try to get access rights of it with several ways. Find the weak points of average web-page's security. Investigate how to prevent loosing control of content changing, stalling or getting the information from web-page's database. Create a simple strategy of protection of web-page that will reduce chances to be hacked by intruder.

The proposed extent of the thesis

30-40 pages

Keywords

database security, web design, password protection,hacking

Recommended information sources

CLARKE, Justin. SQL injection attacks and defense. Waltham, MA: Elsevier, c2012. ISBN 978-1-59749-963-7.

FOSTER, James C. Sockets, shellcome, porting and coding: reverse engineering exploits and tool coding for security professionals. 667 s. ISBN 1-597490-05-9.

GROSSMAN, Jeremiah. XSS attacks: cross-site scripting exploits and defense. Burlington, Mass.: Syngress, c2007. ISBN 978-1-59749-154-9.

CHERRY, Denny a Thomas LAROCK. The basics of digital privacy: simple tools to protect your personal information and your identity online. ISBN 978-0-12-800011-3.

Expected date of thesis defence

2018/19 WS – FEM (February 2019)

The Bachelor Thesis Supervisor

doc. Ing. Vojtěch Merunka, Ph.D.

Supervising department

Department of Information Engineering

Electronic approval: 23. 2. 2018

Ing. Martin Pelikán, Ph.D.

Head of department

Electronic approval: 23. 2. 2018

Ing. Martin Pelikán, Ph.D.

Dean

Prague on 30. 11. 2018

Declaration

I hereby declare to have compiled this final thesis “Database security of web pages” entirely myself and in accordance with recommendations of my supervisor, that I indicate all the literature and other supporting materials used in the index of bibliography.

In Prague 30th of November _____ Ilya Chernookov

Acknowledgements

I would like to thank my supervisor doc. Ing. Vojtěch Merunka, Ph.D. for Your time, instructions and advice that were very helpful and essential during writing of this thesis.

Database security of web pages

Abstract

The main purpose of my bachelor's degree work is to research information about threads and vulnerabilities of web-pages, types of attacks on web-page with intention to steal or alter the information from its database and create a simple strategy of protection of the web-page.

Theoretical part consists of describing the main types of attacks on web-pages, why they are possible and how to prevent them. Also short description of plugins that provides security of web-page from that attacks.

Practical part focuses on describing the plugins I used to protect my web-page, their main functions, attacks they prevents and examples how they prevent the attack.

Key words

Database security, web design, password protection, hacking, web-page, vulnerability, thread

Zabezpečení databází webových stránek

Abstract

Hlavním účelem mé bakalářské práce je výzkum informací o výhrůžkech a zranitelnosti webových stránek, typů útoků na webové stránky s úmyslem ukrást nebo změnit informace z databáze a vytvořit jednoduchou strategii ochrany webových stránek.

Teoretická část se skládá z popisu hlavních typů útoků na webové stránky, proč jsou možné a jak je předcházet. Také krátký popis pluginů, který poskytuje zabezpečení webové stránky z těchto útoků.

Praktická část se zaměřuje na popis pluginů, které jsem použil k ochraně mé webové stránky, jejich hlavních funkcí, útoků, které brání, a příklady, jak zabránit útoku.

Klíčová slova

Zabezpečení databáze, návrh webových stránek, ochrana heslem, hackování, webová stránka, chyba zabezpečení, hrozba

Contents

1. Goals of Bachelor Thesis

The security in today's world is very important for everyone. People want to feel safe and secured in their homes. The internet is enormous world which growing every day with high speed and people which uses the internet wants to feel safe there as well.

The internet is a net that consists from related with each other web-pages. It is growing with increasing web-pages amount. The internet is a parallel world which has it's intruders - hackers.

Hackers are the people that normal users want to be secured from. Normal users do not want hackers to steal their personal data or to change content of their web-pages in internet. Hackers uses the same vulnerabilities to make the same kinds of attacks on users or web-pages always. In theoretical part of this thesis will be reviewed the most frequent kinds of attacks.

The protection must be simple to let a regular users of internet use it.

2. Methodology

The topic consists of this stages:

- Define and describe the most common types of attacks and vulnerabilities
- Describe the reason of origins of each type of attack and vulnerability
- Describe the way to eliminate each type of attack and vulnerability
- Investigate the plugins for protection of the web-page
- Implement the most useful and functional plugins on test web-page
- Show the examples of work in action of each implemented plugin

The most common attack must be described to understand how they work. Find the reason of origins of vulnerabilities and explain how to get rid of them. Search available defense plugins, that are able to defend from described vulnerabilities. Choose the most functional and useful plugins and implement them on test web-page. Collect statistics of attacks occurred during test web-page was online. Show the examples of attacks occurred if it is possible, or initiate test attack.

3. Research, theory

3.1 Introduction

All contemporary webpages are working with some data and requires having database to store the data there. Database is very vulnerable part of any web-page because it contains valuable information inside (e.g. usernames, passwords, personal users' informations). That is why the intruder will always try to get database's content and it is necessary to prevent.

In today's world, security of data in the Internet becomes critical. People use Internet for various reasons: communication, searching the information, shopping, playing games, listening to music, storing the information and so on. Over time the reasons why people spend more time in the Internet increases and the amount of data that people stores increases as well. Almost everyone, who understands the power of the Internet somehow tried to leave his or her mark there. The easiest way to do it is to create your own web-page. With using Hypertext Mark-up Language (HTML) to represent visual content of web-page and Java to make the content dynamic and animated[1](Foster, 2009). But not everyone understands how to keep it safe from changing it's content or getting the important information from it's database by anyone else.

Nowadays it is very important to keep your data safe. The Internet is enormous and over the time, more and more «Hacking for dummies» books releases. If you will try to find such a book, you will successfully find a lot of them. That books will be completely with no price for them. After reading such a book, new «hacker» definitely wishes to try his new skills. No one can say that he will not choose your web-page as firing field. That is why it is important to have a minimal protection to not seem to the intruder as a piece of cake.

3.2 Database

Database is systematically ordered sequence of information, which creates a structure that can be easily processed by computer. The information in database can be various: names, birthdates, credit card information, transaction information and so on[2](Connolly, 1995).

About 15-20 years ago, everyone had a simple database in their home. When the telephones had no phonebook inside, the only way to phone somebody was to enter the number by hand. The number you could get from paper phonebook, a large book with numbers of all telephones in the city. I guess it was a first database that people from my generation faced with. The information that stored in that database was telephone numbers and names and it was stored in an alphabetic order.

The main part of any database is - Database Management System (DBMS). It is a software application, which makes database so comfortable to use. Using it user can create a new database, update the information in existing database, or get access to database for searching a specific information. The user can order the information by various attributes, sort or filter that information when and how he will need it. Wherein it is very important to choose right model of database.

3.3 Common vulnerabilities.

Vulnerability is some weak point in the system, which can cause various problems up to loss of control of the system. Vulnerabilities arises as a result of errors while the programming process, drawbacks that were made while designing of the system. Sometimes is enough for intruder to find just one vulnerability to get full control of a web-page or to get all information from it's database. Not only web-pages suffer from the vulnerabilities, just this year a security researchers revealed two critical vulnerabilities of Intel chips, that were produced since 1995. Those vulnerabilities were named «Meltdown» and «Spectre». The intruder using this vulnerability can get access to an information from running applications, messengers, browsers or emails. Software companies are already released patches, which will prevent intruders to use that vulnerabilities but the chips will be degraded as a result. This will cause the slowing down of home and work computers, as well as cloud services that host popular sites and services. The vulnerabilities will disappear only in new generation chips[3](Chrang84, 2018). The vulnerability is like an opened door, the intruder is just must find it and break in.

3.3.1 SQL-injections.

3.3.1.1 Description of vulnerability.

One of the prevalent type of breaking web-pages is SQL-injection. This type of vulnerability occurred when web pages started to work with databases. Using SQL-injection the intruder is able to get access directly to database of a web-page, can modify the data in the database or simply find an administrator's username and password in the table of users. However, not only the usernames and passwords are usually kept in the database. More information that is valuable is usually stored there, for example personal data of the clients of web-store, their personal phone numbers, e-mails, or sometimes credit card numbers, their expiring dates[4](Clarke, 2009).

The mail thread is lost reputation for the company, whose database was compromised. A few people wishes to use the service, from which someone can pull out any information about them. However, not only the loss of information is terrible, substitution could be even worse.

The intruder is able to put an advertisement usually of his client on any vulnerable web-page. On the other hand, use more elegant way - to substitute the telephone number of victim for telephone number of intruder's client. Finally it will be seen this way: visitor will read the description of services on victim's vulnerable web-page and if he will decide to use that services, he will call to the intruder's clients, because the telephone number was substituted. Usually such modification of web-pages is not noticed soon.

3.3.1.2 Reasons of origins.

Reasons of origins of SQL-injections are low quality filters in query field and sometimes input fields and forms on the web-page. Instead of looking on input value as on some characters API(Application Programming Interface) starts to look on it as on commands.

3.3.1.3 Ways to eliminate.

In spite of the SQL-injection is the most prevalent type of attack on web-pages the protection is quite simple: while the development process it is necessary to pay attention on filtering inputs. While coding a programmer should use MySQL functions like «mysql_real_escape_string» or «mysqli_real_escape_string» that will shield quotes.

You can split your database on several parts and set different level of access to every part. The point is the program or user must have access only to what they need and nothing more.

Use data encrypting. That will protect your data even if the intruder will somehow get access to it[5](phpfaq.ru, 2015).

3.3.2 XSS - Cross Site Scripting.

3.3.2.1 Description of vulnerability.

XSS is a type of attack on web-pages, where the intruder adds a malicious code to the vulnerable web-page, that code due to the web-page is vulnerable becomes a part of a web-page and makes this web-page interact with an intruder's server, when the victim opens modified web-page in his browser.

In this type of attack victims are visitors of web-page. The intruder modifies web-page with a code, which gives a command to browser to send visitor's cookies to intruder's server. In

cookies specific information is kept, sometimes even login and password of modified web-page[6](Cherry, 2014). But if the intruder gets administrator's cookies then he will get access to the control panel of the web-page and it's content.

There are two types of XSS attacks:

Reflected. Reflected XSS attack is more complex, because here psychological knowledge needed. The intruder prepares a link with a script inside. He sends modified link to a victim via e-mail or messenger and here is the most complicated part begins. He must somehow make a victim to do an action: to follow that link, which the intruder prepared to run the script. If the victim won't do an action - the script wont run.

Stored. This way the intruder does not need to lure the victim to follow the links. However, the difficult part in this way is to find a vulnerable web-page or web-resource. The intruder implant malicious script to some executing file on server, usually using forms. This way all visitors of modified web-page becomes victims.

Sometimes the intruder for implementing long-term computer attack on user can use «stored» XSS vulnerability for redirecting users to download Trojan horse. This type of attack called «XSS Based Trojan Horse». The victim loads the vulnerable web-page and the malicious script opens a new tab with zero size, where downloading of the Trojan horse started[7] (Grossman, 2007).

3.3.2.2 Reasons of origins.

The reason of origin of XSS vulnerability is the same as SQL-injection. It is low quality filtering of input values, which helps to intruder to execute logical commands.

3.3.2.3 Ways to eliminate.

To protect your web-page from injecting XSS script is necessary to pay attention on filtering inputs while the development process. Instead of SQL-injection XSS attack is aimed to the users that is why user has to protect himself somehow. Further, I will mention some marks how not to become a victim of XSS attack:

- never follow suspicious links

- do not use cookies(or use session cookies, which deletes after closing tab)

- always be sure that your browser is updated

3.3.3 CSRF - Cross Site Request Forgery.

3.3.3.1 Description of thread.

CSRF is a type of attack on users, leading to the execution on some target web-page of various actions on behalf of registered users on which the attack was made. From target web-page's view the action is made by the user, but from the user's view - there were no action.

This type of attack requires an active session of user on the target web-page. Means the user has to be authorized on the target web-page otherwise the attack won't be committed.

This attack is look like the XSS attack. The intruder prepares some link with a malicious script or he modifies some vulnerable web-page. Sometimes the intruder creates phishing web-page where he can put a malicious script everywhere. For example, he can put the script inside the «src» attribute of <img> tag. The browser will try to load the image, when the user will open the web-page, but instead of loading image it will run the script. That script can be various: from a request to do some spam dispatch to the request to send money to a certain bank account. It depends what the target web-page will be. Depending if the user has an active session on target's web-page, the attack will be committed[8](a.Amirov, 2012).

3.3.3.2 Reasons of origins.

The reason of origins of CSRF attack is the same as XSS attack and SQL-injection that were described above.

3.3.3.3 Ways to eliminate.

The mechanism of protection your web-page against CSRF attacks on Joomla is simple. Server generates a special code named token, which is sent to user for future authentication of inquiry. That token is stored in cookies and have some expiring time. When the user is doing some action on the page, each inquiry sends the token. Server checks the token and if it is up to date, the inquiry will be performed.

To secure your web-page from CSRF attacks also you should follow recommendations below:

- Do not browse other web-pages in the same browser while you are logged into your web-page.
- Log out after you finished working with your web-page
- Do not stay logged into your web-page while you are not doing anything

- Ensure that the address in the browser bar matches the address of your site[9](b.Amirov, 2012).

3.3.4 Brute force attacks.

3.3.4.1 Description of thread.

Brute force attacks is a method of hacking users, by selecting a username and password. The intruder usually creates a program that will consistently sort out all symbol's combinations until your password will not be selected. From the mathematical point of view you can always pick the right password this way, but the time you can spend selecting it can be enormous[10](sfztn.com, 2015).

3.3.4.2 Reasons of origins.

The reason of origins of Brute force is no time out between attempts to enter a password. This allows the intruder to automate the process and use a program. That program can select a password with very fast speed and tirelessly. The only what the intruder has to do is wait and only hardness of the password will set how long he has to wait.

3.3.4.3 Ways to eliminate.

To protect your web-page from a Brute force attacks you have to set a time out between attempts to enter a password. It will greatly increase the time that the program will spend solving your password. If you cannot add a time out (you are not an administrator) you can simply increase the complexity of your password following this advices:

- create long passwords

- use uppercase, lowercase, numbers and special symbols in your password

- never create password the same as your username

- creating a password do not use your personal information

- for every account create unique password

- regularly, at least monthly, change your password[11](Ladypain, 2018).

3.4 Database protection.

It is very simple to make a web-page using Content Management System(CMS), but it is not very safety. This means that your web-page is vulnerable for the intruders, because basically

all CMS are open source. The intruder can read a code of CMS, can find a vulnerability and use it to get access to the admin panel of your web page and after he will be able to do anything with content of a web page and data inside the database.

To prevent such a scenario you should use a protecting plugin or combination of them. Further will be described some protecting plugins for the popular CMS Joomla.

3.4.1 Admin Tools

Admin Tools is a protecting plugin, which provides a complex defense of the web-page. It protects web-page against SQL-injections, optimizes database's tables, let the most useful maintenance operations run automatically such as file change scanner. In case of an attack, the plugin can put the web-page off-line. Allows one-click temporary files cleaning or setting the time of automatic cleaning.

3.4.2 CMS-Security

CMS-Security is a component that provides some security features to the web-page. It provides a Firewall, that every modern web-page needs to protect the sensitive data and users. Advanced dashboard provides an informative security and firewall options. Simple checklists helps to find security or firewall issues on the web-page and quickly sort them out with one click. This extension provides an informative website security informations including:

- Informative dashboard
- Social media information
- Security and firewall checks
- Administrator tasks
- Website checks options
- File and Folder permissions
- Black and White IP's with search options
- Emergency shutdown

3.4.3 RSFirewall

This is one more plugin with a complex defense for web-pages. The feature of this plugin is protection from the list if an injections, informative dashboard with a list of attacks,

automatically adding to blacklist of an attacker's IP, allows user to select which countries won't have an access to the web-page, notifies if the sensitive Joomla files were changed, and so on.

3.4.4 AdminExile

This plugin allows you to protect your admin panel from the intruders. It changes the link that links to login form of the admin panel. With a several options - use access key + key value or access key only - the user can complicate the access to admin panel of the web-page.

3.4.5 Marco's SQL Injection

A simple plugin that provides a protection against SQL-injections and Local Files Inclusion attacks.

4. Implementation

I created an informational web-page for an international organization. This organization is making excursions on different languages. Visitors must have an access from any country. The auditory is multinational and that is why this web-page requires a protection.

For my web page, I have chosen these three plugins: RSFirewall, AdminExile and Marco's SQL Injections. It is necessary to use a mix of plugins to let them overlap limitations of each other. Further, I will describe chosen plugins deeply.

4.1 RSFirewall

RSFirewall is a protecting plugin whose main task is to protect the webpage from any kind of injections. His main capabilities include:

- Real-time protection from SQL-injections, PHP, LFI, XSS attacks and injections
- Protection of admin panel by password and additional layer of protection
- Web-page change blocking
- Analyzing web-page's database
- Advanced system log
- Access blocking to the settings of RSFirewall with a password
- IP-address black list, who are not allowed to access to the web-page
- Distinction of access rights for administrators to the admin panel and different components of the web-page
- Basic protection from DoS attacks
- Uploading files filtering
- Protection of scanning web-page for vulnerabilities
- Protection of administrator's account from changing
- Basic protection from Brute Force attacks
- Spam protection for forms

RSFirewall is available in 14 different languages: Arabic, Brazilian Portuguese, Danish, Dutch, French, German, Indonesian, Italian, Polish, Russian, Spanish, Swedish, UK English, and US English.

Advanced system log allows seeing the information about recent system activities. For example, the information about last enter to the system shows. It includes level of danger(low, medium, high and critical), date and time, IP address of the user, the name of his account, description of the event(e.g. Entered the admin panel with right password), and the link to the RSjoomla web-page with detailed information of the event.

RSFirewall distinguishes four levels of danger:

- Low. (e.g. Access to RSFirewall with right password)
- Medium. (e.g. Access to RSFirewall with wrong password)
- High. (e.g. Disabling the system lock)
- Critical. (e.g. Changing file *configuration.php*)

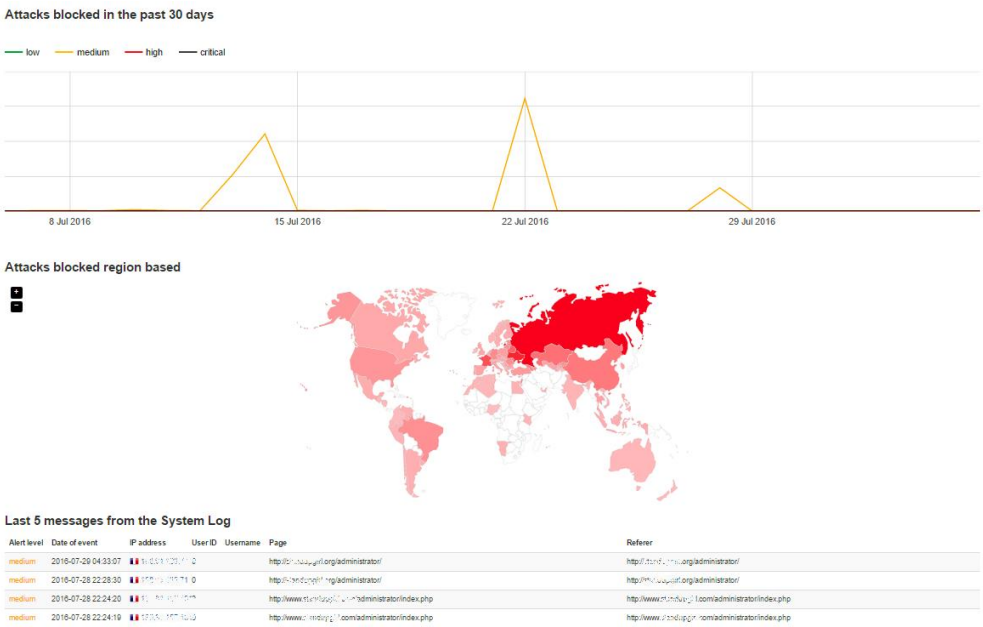
Against Brute force attacks, the user can set up maximum value of failed access attempts to the administrator panel. After exceeding this value automatically, CAPTCHA will be activated. This will protect your web-page after automatically selecting login and password.

RSFirewall allows you to lock the system. After locking the system, it becomes impossible to install, update or delete the extensions i.e it locking access to *com_installer*. Also all new created users with administrator's rights will be deleted automatically. The locking does not affect the registration of new users, adding, changing and deleting materials, menu items, modules, categories, changing the basic settings of Joomla and so on.

4.1.1 Example

The system overview page consist of graph of prevented attacks on your web-page for the last 30 days, the world map with countries from which the attacks were occurred and a table with detailed information about that attacks. The map with shades of red shows you from which country were the most count of attacks occurred, and the darker color is the more count of attacks were made. The attacks have a level of danger, as it was mentioned in description, and distinguished by the color.

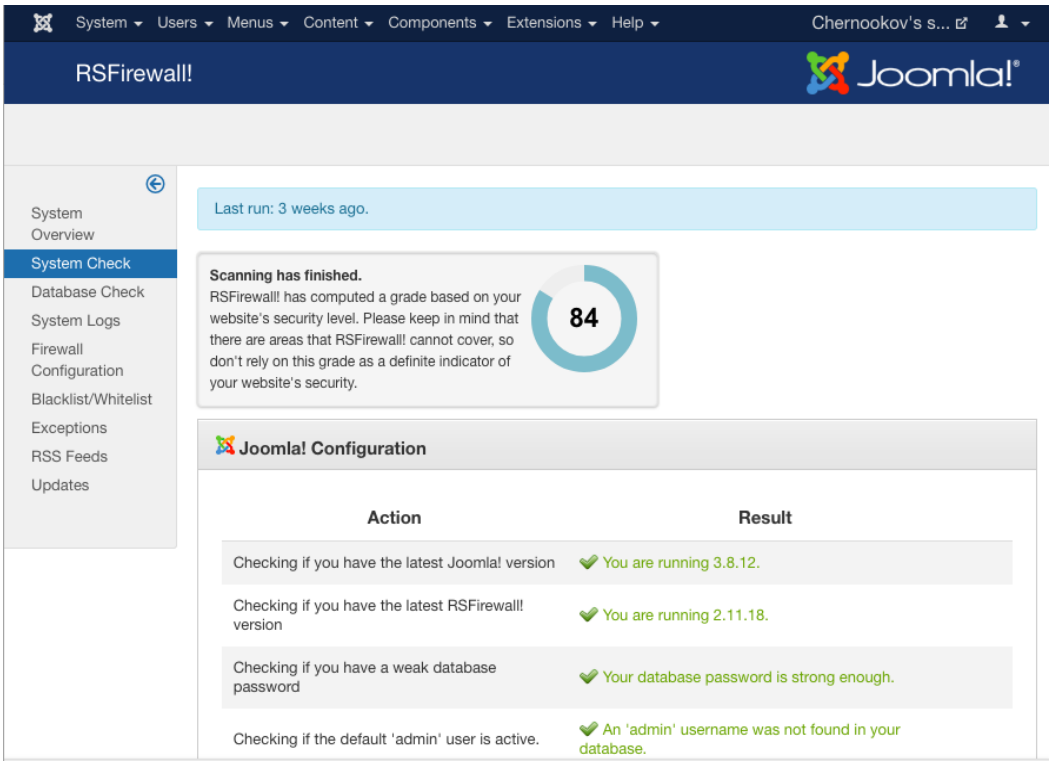
Figure 1: RSFirewalls System Overview



Source: https://www.rsjoomla.com/support/documentation/rsfirewall-user-guide/getting-started/system-overview.html

RSFirewall allows you to run a system check.

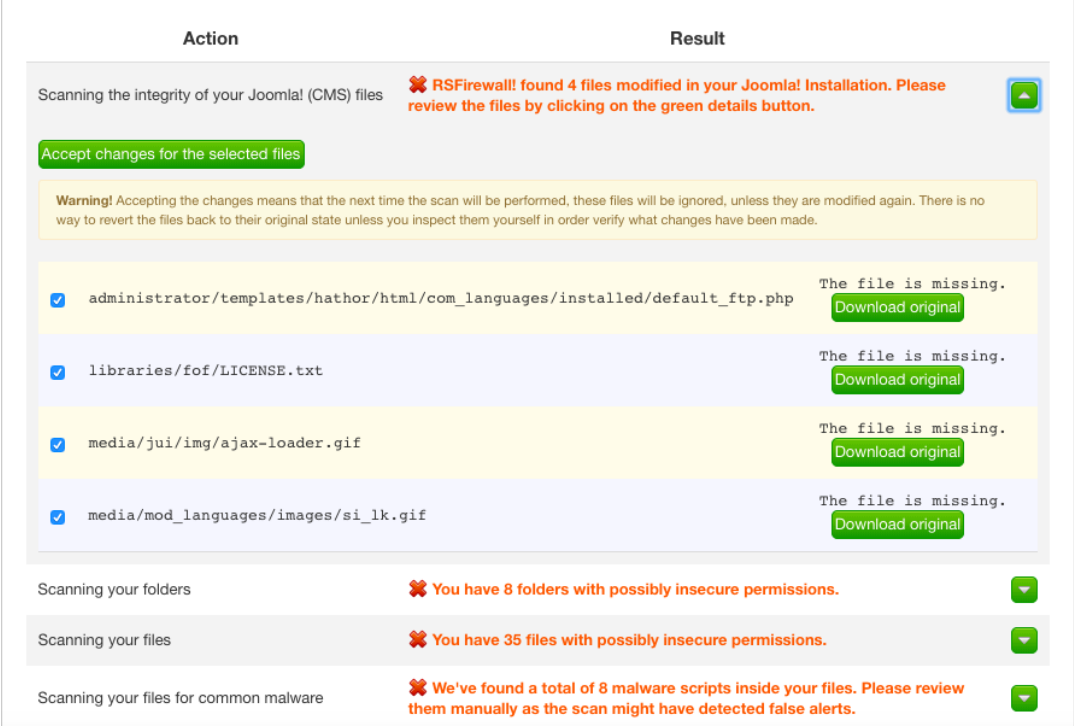
Figure 2: RSFirewalls System Check



Source: Author

Plugin checks Joomla configuration, server configuration and file integrity for vulnerabilities or errors. After performing the system check, the plugin gives some points to your web-page according to the number of vulnerabilities it has found.

Figure 3: RSFirewall plugin's File Integrity

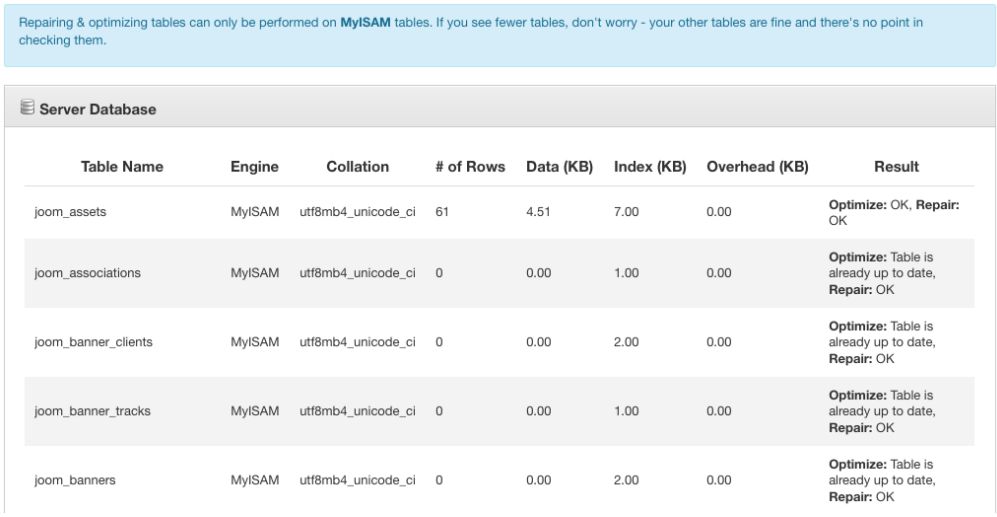


Source: Author

The file integrity scanning checks if the core files of Joomla have been altered. The plugin compares the actual file with a pre-calculated hash of the original core file. This tool is not only detection tool it also tries to repair the detected problem. If it finds out that the files were altered it allows you to view the differences of the modified file and to download original file.

The database check allows you to see if any of your databases were corrupted.

Figure 4: RSFirewall plugin's Database Check



Source: Author

The plugin checks, analyzes and optimizes the tables of your web-page. If any of the table is found corrupted, the plugin will try to repair it without losing data from the table.

4.2 AdminExile

One more interesting protection plugin is AdminExile. The main feature of this plugin is that it changes the link to the administrator panel. After installing this plugin it will not be possible to enter the administrator panel entrance with just adding */admin* or */administrator* to the end of URL of your web-page. That is actually very useful feature, because it creates an additional layer of defense, without overcoming that the intruder could not use Brute Force attack to your web-page.

AdminExile allows you to set the Access Key and Key Value and your URL becomes something like this: http://your-web-page//administrator/index.php?*accesskey=keyvalue.* If the user will not enter right Access Key or Key Value, he will be automatically redirected to the home page of your web-site. You can manually set the link where the user will be redirected in settings of AdminExile. It is possible to set up the time of re-entry, the time when you can come back to administrator panel without entering the Access Key and Key Value, when for example you accidentally closed the tab.

If you forget your Access Key and Key Value, it is possible to enable sending an e-mail with the link to the administrator panel's entrance after entering to the query a special command. The groups where the e-mail will be sent you should set beforehand.

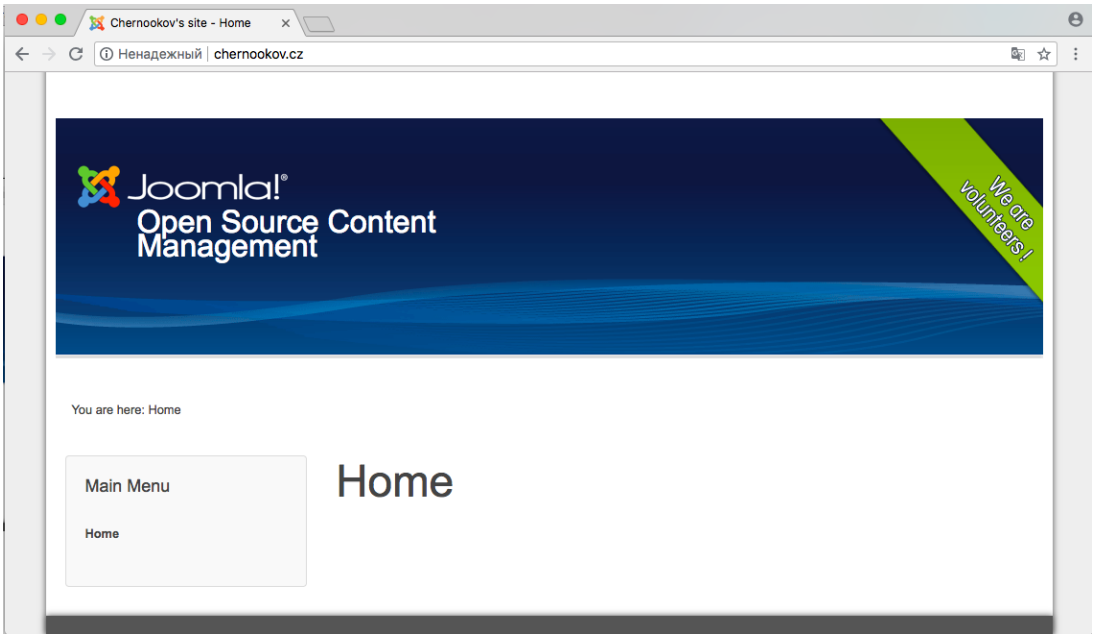
Besides the main feature, AdminExile also has an IP-address black list and white list, where you can block some IP addresses or set the only IP-addresses from which you can log in to the administrator panel.

Simple Brute Force defense in case the intruder gets your link or somehow recognizes your Access Key and Key Value. Here you set up the number of failed access attempts, time penalty when the intruder cannot re-enter his password if the number of failed access attempts will be exceeded and the penalty multiplier, it is the number that will be multiplied time penalty after every future failed access attempt.

4.2.1 Example

Let's imagine us as an intruder, we created a special program that allows us to use Brute Force to hack a web-page. We noticed a simple web-page on Joomla with an address www.chernookov.cz and this web-page is seems to be a very good polygon to test our program.

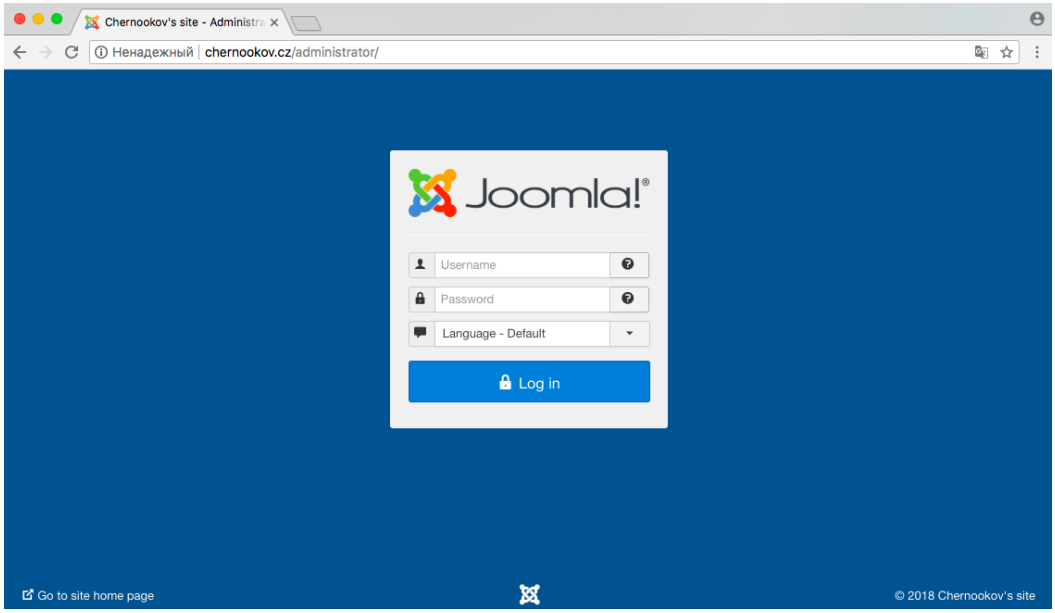
Figure 5: Test web-page's home page



Source: Author

To use our program we need to find any login form, as we see on the picture above on this web-page regular users does not have a possibility to log in. All Joomla's web-pages have an admin panel and to reach it administrator must add /administrator to the query and use his login and password.

Figure 6: Test web-page's authorization panel

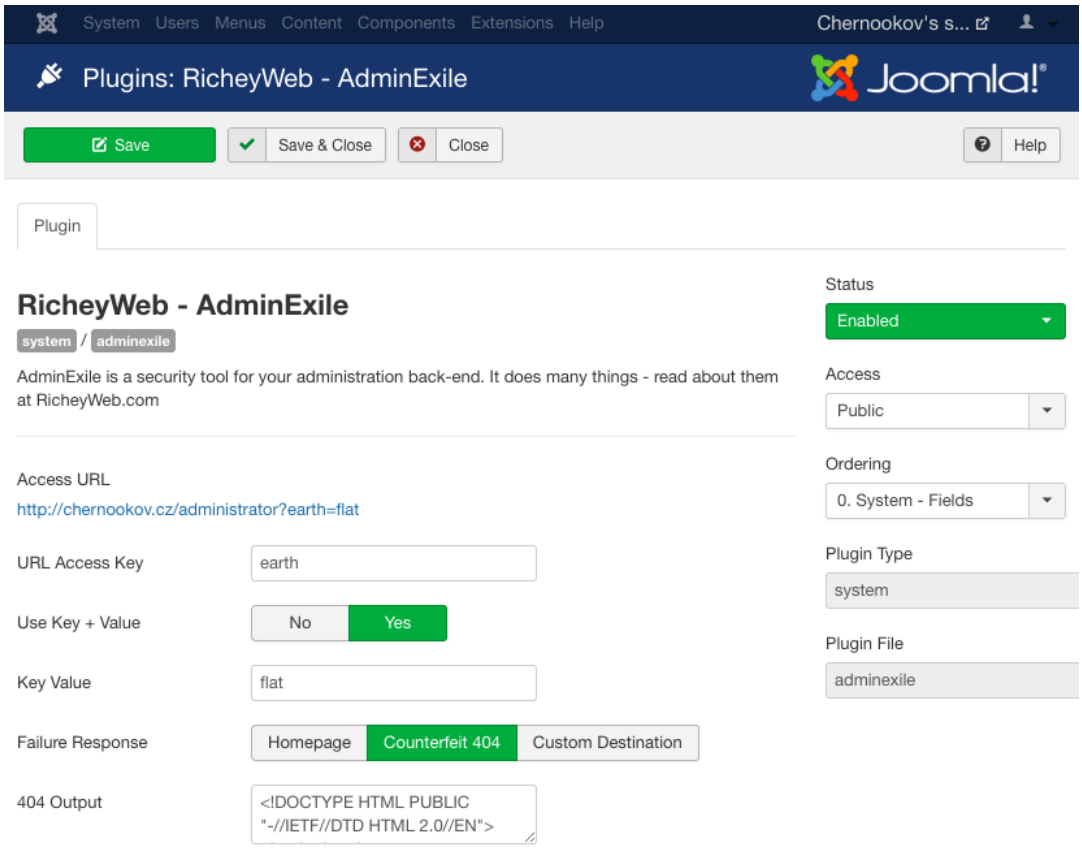


Source: Author

After adding /administrator to the query, we see a log in form. That is exactly what we need to test our Brute Force program.

Simple enabling the AdminExile plugin and adding Access Key and Key Value preventing this scenario.

Figure 7: AdminExile plugin's configures

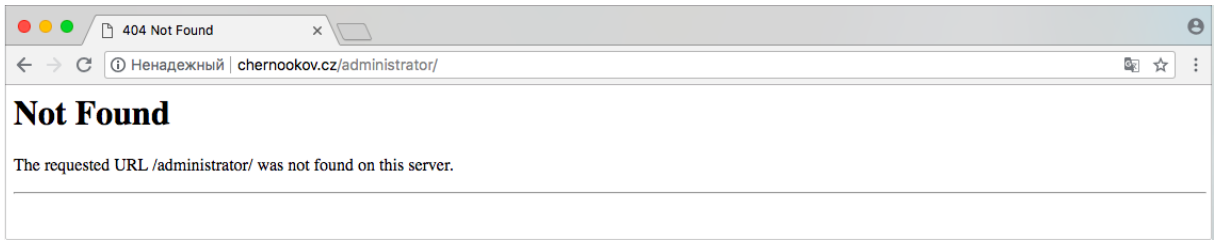


Source: Author

As we see in our configurations, the Access Key and Key Value are set and the plugin's status is enabled.

Now let's try to go to login form again.

Figure 8: Hidden authorization panel



Source: Author

Here we see our protection in action. As you may notice the query contain the same address as previous picture, but after enabling the protecting plugin, it is no longer enough to see the login form.

4.3 Marco's SQL Injection

Marco's injection interceptor will protect your web-page from most of SQL injections. This plugin adds a simple but, in most cases, fundamental protection against SQL injections and Local Files Inclusion(LFI) attacks. It checks files and data that were sent to Joomla, intercepts a lot of common exploits and saver your web-page from intruders.

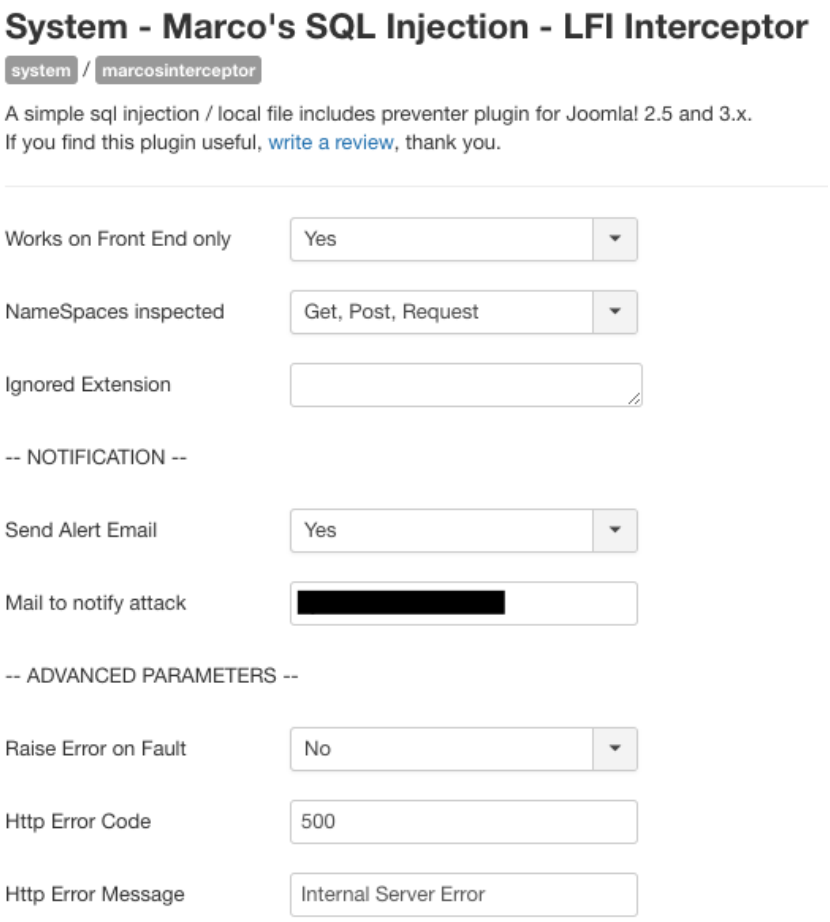
Its main functions:

- Blocks SQL injections
- Blocks LFI attacks
- Filters POST, GET, REQUEST requests
- Notifies you by e-mail when attack comes
- Include white list for safe components
- Automatic attacker's IP blocking
- Protect from 3rd party extension's vulnerabilities

4.3.1 Example

On the picture below, you can see configurations of this plugin I used to protect my web-page.

Figure 9: Marco's SQL Injection plugin's configurations

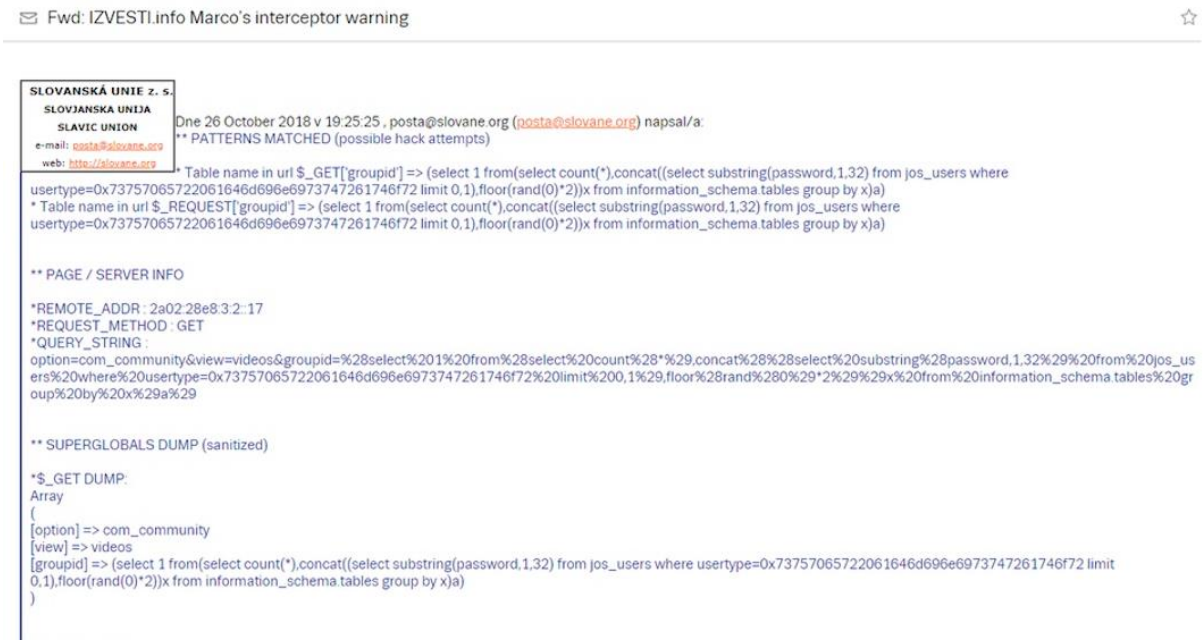


Source: Author

Most of the configuration I left as it was set default. There were only few configured features:

- Works on front end only – Here you set "Yes" if you want to intercept injections only on front-end page only, means your admin panel stays unsecured. In my case, my admin panel is hidden by AdminExile plugin, so I can left this configuration as default.
- Mail to notify attack – enter an e-mail where you want to get mails about attacks on your web-page.
- Raise Error on Fault – If you set "Yes" here, after every failed injection the error message will appear. I set here "No" because I think that it is unnecessary to inform an intruder.

Figure 10: Marco's SQL Injection plugin's example of notification



Source: Author

On the picture above you can see an example of notification about an attack on your web-page. There you can find various information about attack like method that was used or query itself. Usually it is nothing to worry about when you get such a notification, because plugin works automatically and if you gets one you can relax, because it means that your plugin is working properly.

5. Discussion

My web-page that I used to help me to complete this thesis was online for 3 month in a row. For this period over 300 attacks were committed. The web-page was secured with plugins, I described in practical part.

33.8% - SQL-injections. Total: 103 attacks.

28.5% - XSS-attacks. Total: 87 attacks.

19% - LFI attacks. Total: 58 attacks.

18.7% - CSRF attacks. Total: 57 attacks.

0% - Brute force attacks. Total: 0 attacks.

As you can see above, I mentioned total amount of each attack that was committed. The first one is SQL-injection, it got the biggest amount of attacks on my web-page. Two out of three plugins, I installed are able to protect the web-page from SQL-injections: RSFirewall and Marco's SQL Injection.

Right after SQL-injections are XSS-attacks with almost the same total amount of attacks. The main defender from this type of attack is RSFirewall, but also AdminExile can prevent reflected XSS-attack by changing the link to admin panel.

LFI attacks are on the third place. From this type of attack protects RSFirewall and Marco's SQL Injection.

The least popular type of attack was CSRF attacks, this type of attack is completely the same as XSS attacks, and the same plugins protected my web-page: RSFirewall and AdminExile.

As you noticed, there was not any Brute force attacks on my web-page. The reason is the protection by AdminExile, that prevents seeing input forms for logging in and prevents the opportunity to use this type of attack on my web-page.

6. Conclusion

After study of suitable literature, I was able to review the description of main dangers of the internet for the users and their web-pages in this work. Described reasons of origins of each dander and gave some advises how to avoid them.

Internet is a very big world of web-pages. Like our world, it has his own dangers and traps. To avoid becoming a victim of a trap the user have to be awared of them. The users are cultivating the internet, creating more and more web-pages for various reasons every day. Users, like parents wants to protect their children, wants to protect their web-pages from the intruders and today's Content Management Systems allows them to do it automatically.

In practical part, I used my web page to show the defense I used to protect it. The installation of plugins is very simple and every regular user will be able to install them. Every plugin is working automatically, and it is enough just to install it to make your web-page more safe. Some of them requires some configurations, I described in practical part, which I used for my web-page.

The intruder will dictate the future of defense in the internet. Because the intruder uses all his imagination to break through the defense and to reach his aim. However the defense always will be evolving to do it's function.

7. Bibliography

1. FOSTER, James C. Sockets, shellcome, porting and coding: reverse engineering exploits and tool coding for security professionals. 667 s. ISBN 1-597490-05-9.

2. CONNOLLY, Thomas M a Carolyn E BEGG. Database systems: a practical approach to design, implementation, and management. 5th ed. Boston: Pearson Education International, c2010. ISBN 978-0-321-60110-0.

3. Meltdown and Spectre [online]. [cit. 2018-10-13]. Avaliable from: https://habr.com/post/346164/

4. SQL injection attacks and defense. Burlington, MA: Syngress Pub., c2009. ISBN 978-1-59749-424-3.

5. SQL injection protection [online]. [cit. 2018-10-13]. Available from: http://phpfaq.ru/mysql/slashes

6. CHERRY, Denny a Thomas LAROCK. The basics of digital privacy: simple tools to protect your personal information and your identity online. ISBN 978-0-12-800011-3.

7. GROSSMAN, Jeremiah. XSS attacks: cross-site scripting exploits and defense. Burlington, Mass.: Syngress, c2007. ISBN 978-1-59749-154-9.

8. Vulnerability CSRF. Introduction [online]. [cit. 2018-10-13]. Available from: https://intsystem.org/security/learn-about-csrf-intro/

9. Vulnerability CSRF. Protection [online]. [cit. 2018-10-13]. Available from: https://intsystem.org/security/learn-about-csrf-security/

10. Brute force – Password collection [online]. [cit. 2018-10-13]. Available from: https://sfztn.com/security/brutfors-podbor-parolya-pereborom

11. Brute force – is... Description of the program, installation procedure, protection [online]. [cit. 2018-10-13]. Available from: http://fb.ru/article/428987/brutfors---eto-opisanie-programmyi-poryadok-ustanovki-zaschita

8. Table of figures