Possibilities of conducting XSS-attacks and the development of countermeasures

Yelena Revyakina^{1,}, Larissa Cherckesova¹, Olga Safaryan¹, Denis Korochentsev¹, Nikolay Boldyrikhin¹, Yuri Ivanov²*

¹ Don state technical University Rostov-on-Don, Gagarin square, 1, Russia

² Southern Federal University Taganrog, 44, per. Nekrasovsky, Taganrog, Rostov region, Russia

Abstract. The article describes the investigation process of the possibilities of XSS-attacks, and the development of counteraction means to these attacks. Researches were determined whether XSS-attack can be fulfilled successfully, and vulnerability detection methods can be applied; were developed the logical and structural diagrams of XSS-vulnerability detection program; were realized program implementation (software) of algorithms for detecting XSS-vulnerabilities on the Web – sites. The software implementation is Web extension for the Google Chrome browser. Main purpose of implementing this software is to confirm or deny the presence of XSS-vulnerabilities on the site, and to counteract the possible attack.

1 Introduction

Today, the informatization is one of the priority directions for the development of all economic sectors. Almost any business, commercial or government, has its own Web-site and offers various online services. Personal data of clients and employees, financial information, and business activity data are stored electronically. The huge volume of compromised personal data and payment information, and the high percentage of leaks of these types of data, indicates the growing value of personal information in digital form from year to year.

Moreover, this applies not only to personal data of individual citizens, but also to information about individuals – representatives of counterparty organizations, which personal information is accumulated in the client databases of the commercial companies. In this regard, the task of ensuring the security of Web-applications is becoming more and more relevant at the moment and is a priority for a huge number of companies whose activities are connected with the Internet.

Unfortunately, developers of corporate information systems do not always follow security requirements — due to lack of necessary experience or simply focusing on other goals when developing the system. At the current moment, huge number of different Web-applications have such vulnerabilities, which can cause the organization to suffer financial or reputational damage. The most common and dangerous vulnerabilities, according to “Positive Technologies” magazine, is XSS-vulnerabilities (Cross Site Scripting).

*Corresponding author: revyelena@yandex.ru

The term is abbreviated "XSS" to avoid confusion with cascading style sheets that use the abbreviation "CSS" [1].

This vulnerability in 2019 had 77 % Web-applications. Of all attacks on the Web-applications recorded in 2019, 14% of attacks related to the XSS-attacks type [2].

It is not possible to ensure appropriate level of application security without conducting security testing. However, conducting such testing manually involves enough large amount of resources and requires more qualified developers, and sometimes – special department responsible for security.

In addition, the human factor in testing application security can play the disastrous role for the company. Therefore, it is recommended using automated testing to ensure an appropriate level of security [3]. When creating new system, it is needed to rely on the experience of existing software systems that have similar functionality, as well as take into account the main pros and cons of these systems [4], [5].

XSS – attacks is specific type of attack on Web-systems that involves inserting malicious code into the page issued by the Web-system (which will be executed on the user's computer when they open this page) and interacting with the attacker's Web-server (figure 1). It is type of "code injection" attack [6].

Fig. 1. XSS action algorithm.

The specific feature of such attacks is that malicious code can use the user's authorization in the Web-system to get extended access to it or to get the user's authorization data. Malicious code can be inserted into the page either through vulnerability in the Web-server or through the vulnerability on the user's computer.

The object of study is Web – pages with XSS vulnerabilities.

The subject of research is attacks on Web – pages.

The scientific novelty of this work is studying the possible XSS-vulnerabilities on Web – pages; estimation the possibilities for attackers to conduct XSS – attacks when they find such vulnerabilities; and development of measures and tools for counteraction to attacks when creating Web – extension for the Google Chrome browser software for checking and testing Web – pages for XSS vulnerabilities.

Purpose of investigation is creation of program application for Google Chrome browser, of automated system for testing Web – pages for XSS vulnerabilities, with understandable intuitively interface, easy configuration and high percentage of vulnerability detection. In accordance with this supplied goal, the following tasks were defined for its achievement: analyze the subject area; examine existing software tools for detecting XSS– vulnerabilities, identify advantages and disadvantages; determine the possibilities of detection methods application of XSS vulnerability for the attacks preventing; develop the logical and structural

diagram of the XSS – vulnerability detection program application; – realize the software of the algorithm for detecting XSS vulnerabilities on Web-sites for Google Chrome browser.

2 XSS – Attacks Types and Methods of their Detecting

Reflected cross-site scenarios (XSS) occur when attacker inserts executable browser code into an HTTP response. The embedded attack is not stored in the app itself, is not permanent, and only affects users who open the created malicious link or third – party Web – page [7].

One of the main difficulties in the preventing XSS vulnerabilities is proper character encoding. In some cases, the Web-server or Web-application cannot filter the certain character encodings, so for example, the Web-application can filter "<script>", but cannot filter % 3cscript% 3e, which just includes the different tag encoding [8].

Thus, most of the prevention of XSS-attacks should be based on building Web-application filter mechanisms against unreliable user input. Developers can use several mechanisms, such as returning an error, deleting, encoding, or replacing incorrect input. The means by which the application detects and corrects the results of incorrect input is another major drawback in preventing XSS-attacks [9].

Stored Cross-Site vulnerabilities (XSS) are the most dangerous type of Cross-Site Scenarios. Web-applications that allow users to store data are potentially susceptible to this type of attack. Stored XSS is particularly dangerous in application areas where users with high privileges have access. When administrator visits vulnerable page, their browser automatically executes the attack. This can expose sensitive information, such as session authorization tokens.

At the current moment, XSS-vulnerabilities are in third place in the ranking of key risks of Web-applications according to OWASP rating (Open Web-Application Security Project). There are several types of XSS-vulnerabilities, divided by attack vector, script implementation channels, and method of exposure.

There are 2 main ways of protection against XSS vulnerabilities: server-side protection and client-side protection. Today, there are many different tools for detecting XSS-vulnerabilities, including NetSparker, XSS-Me, and Wapiti.

Expansion (browser extension) is additional module that can be connected to the browser to add certain functions. Depending on the browser, the term may differ from the symbols of set designations, for example, plug-in, add-on, or extension. Expansions can change the browser interface or add features for the comfortable working on the Internet.

For example, it can be used extensions to block the advertising (marketing), translate pages from foreign languages, or add page addresses to special bookmark services like Evernote or Pocket.

There are lot of extensions – thousands and thousands of different programs for every taste: for more convenient and efficient operation, for configuring the browser's appearance, for online purchases, games, getting cashback, and much more. Extensions support almost all popular browsers: Chrome and Chromium, Firefox, Safari, Opera, Internet Explorer and Edge. Extensions are easy to download, and can be very useful, so many people use even several extensions, and sometimes several dozen of extensions [9], [10].

There are several types of extensions [11]:

1) Integration of the extension. Cloud software for Customer Relationship Management systems (CRM). It is software for organizations designed to automate strategies for interacting with customers, in particular to increase sales, optimize marketing and improve customer service by storing information about customers and their relationship history, establishing and improving business processes, and then analyzing the results. Warehouse management programs, and many other business management systems is becoming more

popular, and no cloud provider gives to user the opportunity to integrate into them and fit them to their own needs.

2) Microservices. These are micro applications that are called "on the button" and interact with the page. They are needed for services that make sense to run in the environment of another site / application. They are able to read information from the site where it was called, and thus significantly simplify the interaction with users.

3) Tools for the developer. There are a huge number of extensions that allow calling various tools for the Web– development without closing browser, which is undoubtedly very practically and conveniently.

Advantages of extensions: quick access, ease and clarity of use; cross platform – the ability to work on any platform where there is browser; ability to integrate the non–integrated, insert its functionality into third–party products in the core of which there is no access; the ability to combine systems and cloud services into comprehensive corporate system landscape.

Disadvantages of extensions: need to periodically update the extension for browser or service updates; you need to write a separate version of the extension for each browser. The SOFTWARE also implements working with the MySQL database via the phpMyAdmin application [12].

3 Development of Software Tool for XSS–Attacks Preventing and Countering

The software tool is designed to check sites for reflected XSS vulnerabilities. Structure of the performed functions [13]:

- 1) Automatic site check for reflected XSS–attacks;
- 2) Automatic information saving about the site being checked in the database, where all data about XSS–attacks is located;
- 3) Notifying the user about the XS–attack conducting;
- 4) Automatic comparison of the site being checked with sites located in the database, and output of comparison results.

The software package consists of several program files [14]:

manifest.a json – file that stores instructions for the browser, application description, and paths to the visualization image;

index.js – file containing application main code and functions;

jquery.js – library for interacting with the document's DOM structure, where Document Object Model (DOM) is platform– and language–independent programming interface that allows programs and scripts to access content of HTML, XHTML and XML documents, to change their content, structure, layout, etc.

The software tool implemented in this research is extension.

Next, look at how it works in detail. The user goes to the site and launches the extension. The extension makes request to the API class. The API class passes control to the BD class and reads information from the database for the current site. If this site is present in database, the status of previous check is returned. If site was not checked, then all forms are downloaded from the page, action attributes are stored, then fields with the text value are selected and their name is saved [15].

After that, pairs of values (action, name) are sent to Web– server, which passes control to XSS–class. It returns either 0 or 1. The general scheme of the program is shown in figure 2.

Fig. 2. Schematic representation of the extension's algorithm.

The program starts working from the moment the user clicks on the site. After the user logs in to the Web–page, the extension first checks whether a vulnerability search has been performed previously. If not, the check will start in the background. Figure 3 shows this algorithm.

Fig. 3. Flow diagram of the extension algorithm.

If extension starts page scanning for XSS–vulnerabilities, then the following procedure is performed [16]:

- 1) First, elements with the <input> tag are selected.
- 2) Selected values read off the name attribute, and identify the parent form element. Action attribute is read for this element.
- 3) Two values of <name of action> is sent to Web–server.
- 4) On Web–server, the URLS addresses of the following format are generated for each data set:

- <<host name> / <current URL> ? <Name> = <XSS insert>
<host name> / <action> ? <Name> = <xss insert>
- 5) After forming, the page with received URLS is loaded and the search for XSS insert on the current page begins working.
- 6) The verification result is saved, and sent to the client.

4 Description of Developed Extension for Preventing XSS – Attacks

The main purpose of implementing this software tool was to confirm or deny the presence of an XSS–vulnerability on the site [17]. The software implementation is Web–extension for the Google Chrome browser.

After it is installed and activated, constant background check of visited Web–pages begins. If a vulnerability is found on a Web–page, the program will warn the user about it.

To install an extension, open Google Chrome browser, go to Settings, and select the Extensions tab. To install third–party extensions, enable developer mode [18].

After that, click on the "Download unpacked extension" button and specify the folder with the program. If the program "XSS checker 1.0.1" appears in the list of extensions, then the installation can be considered successful [19].

After installation, the indicator will also appear in the lower–left corner of the Web – page. If the indicator turns green, it means that there are no XSS vulnerabilities, detected on the page. If the color is red, it means that there are vulnerabilities [20].

After pressing the F12 key, the console will display a list of URLS with possible vulnerabilities. When you click on a vulnerable link, a pop-up window may appear confirming its vulnerability. These procedures are shown in figure 4.

Fig. 4. Viewing the code of page with XSS compatibility.

All information about verified and tested sites will be saved in the database. To view it, it is need to visit the Web– resource Vladimir–xss.space to be created.

It will need to save all the sites visited by the user and those sites where XSS–vulnerabilities are detected. They will be noted, and highlighted in red color [21]. There will also be the button for deleting the site from the database also. The database fragment is shown in figure 5.

XSS

192.168.0.106	No vulnerability	Delete
192.168.0.106	No vulnerability	Delete
192.168.0.106	No vulnerability	Delete
192.168.0.106	No vulnerability	Delete
192.168.0.106	No vulnerability	Delete
esia.gosuslugi.ru	No vulnerability	Delete
elmed-rostov.ru	No vulnerability	Delete
polihrom.com	Vulnerability !!!	Delete !!!
google.ru	No vulnerability	Delete
192.168.0.106	No vulnerability	Delete

Fig. 5. The portion of the database.

5 Conclusion

As result of research, the extension was created using the languages of Web programming – PHP and Javascript, for Google Chrome browser, to check Web – sites in background for reflected XSS – attacks. This application is easy to install and does not require the additional settings. For more clarity, when working with application, it is added site verification indicator and alert if XSS – vulnerability was detected on the site.

The developed application also allows saving results to database, and notifying the user of threat if the current site was checked before it. Information from the database is output to the Web – page that is implemented using HTML and Cascading Style Sheets (CSS) [22] – [25].

In the future, the application can be adapted to other browsers (Mozilla Firefox, Opera, and Edge) and expand the functionality of the program, for example, add the ability to detect stored XSS–attacks and DOM – based attacks.

Fundamental difference of presented elaboration from other developments is investigation the opportunity of presence XSS–vulnerabilities on the Web – pages and studying the hackers’ ability to conduct XSS–attacks. In the work the measures and tools to prevent and counteract to the attacks are developed. Web–extension for Google Chrome browser software to check and test Web – pages for XSS vulnerabilities is created.

Comparison with analogues is quite difficult. There are well–known utilities such as Xenotix, XSSF, or BeFF, but this software is not oriented for Google Chrome browser. It is possible that analogues were developed for other browsers. It is not possible to compare such commercial projects with extensions created by authors, since these analogues are, in any case, paid licensed software with large functionality developed by major team of programmers. However, authors hope that further development of their import–substituting

and free software extension will allow them to compete with existing foreign developments in this area.

The developed application has been tested against non –permanent (reflected) types of XSS–attacks. Developed by authors Web–extension for Google Chrome browser software finds vulnerabilities and reflects certain types of attacks, but does not perform comprehensive complex scanning.

Further development of elaborated application involves adding the capabilities for detecting the stored XSS–attacks and DOM–based attacks, so the authors are planning to test it for detecting other types of XSS–attacks: permanent (stored), and local (based on the DOM model) as well.

The probability of getting false – positive error when software extension testing usually lies in the range of 5 %, but in some cases, it can reach up to 10 %. If there is high probability of receiving the false – positive error, repeated testing is performed, which determines whether the XSS attack is detected correctly.

References

1. *Types of XS: Stored XSS, Reflected XSS and DOM–Based XSS* <https://www.acunetix.com/Website security/xss/>
2. Laponina O R 2016 *Cryptographic Foundations of Security* (M.: National Open University INTUIT) p 242
3. Philips D A, Davis, M E 2019 *Studying PHP and MySQL* (K: "Symbol") p 448
4. Elhadi A M *Complete Guide to Intersite Scripting* <https://www.securitylab.ru/analytics/432835.php?R=1>
5. *Positive Research Vulnerabilities of Web–Applications: It's Time to Analyze the Source Code* <http://www.blogptsecurity.ru/2017/08/Web-attacks.html>
6. Esposito D 2017 *Development of Modern Web–Applications Analysis of Subject Areas and Technologies* (K: "Williams") p 464
7. Gupta S and Gupta B 2017 *International Journal of System Assurance Engineering and Management* **8** 512 – 530
8. Hydara I et al 2015 *Information and Software Technology* **58** 170 – 186
9. Rohilla M, Kumar R and Gopal G 2016 *International Journal of Advanced Research in Computer Science and Software Engineering (IJARCSSE)* **6(6)** 273–279
10. Tomohisa I and Sakurai K 2017 *Proceedings of the 11th International Conference on Ubiquitous Information Management and Communication ACM*
11. Parvez M, Zavarsky P and Khoury N 2015 *10th International Conference for Internet Technology and Secured Transactions (ICITST–2015)* Doi: 101109/ICITST20157412085
12. Shahriar H and Zulkernine M 2011 *Computer Software and Applications Conference Workshops (COMPSACW) IEEE 35th Annual* 104 – 109
13. Shar L K, Beng H and Tan K 2011 *Information and Software Technology* **54** 467–478
14. Parameshwaran I, Budianto E and Shinde Sh 2015 *10th Joint Meeting on Foundations of Software Engineering* 946 – 949
15. Gupta Sh and Gupta B 2016 *ICETEST–2015 Proceedings Technology Elsevier* **24** 1595–1602
16. Yusof A S and Pathan K 2014 *Proceedings 5th IEEE Conf Information and Communication Technology (ICT4M 14)* 1–6

17. Shar L K and Tan H-B K 2012 *Computer* **45(3)** 55–62
18. Kirda E et al 2006 *Proc 21st Ann ACM Symposium Applied Computing (SAC06)* 330–337
19. Jim T Swamy N and Hicks M 2007 *Proc 16th Int'l ACM Conf World Wide Web (WWW07)* 601–610
20. Nadji Y, Saxena P, and Song D 2009 *Proc 6th Ann Network & Distributed System Security Symposium (NDSS09)* <http://www.cs.berkeley.edu/~dawnsong/papers/2009%20dsi-ndss09.pdf>
21. Louw M T and Venkatakrishnan V N 2009 *Proc 30th IEEE Symposium Security and Privacy (S&P09)* 331–346
22. Thankachan A Ramakrishnan R and Kalaiarasi M 2014 *Information Communication and Embedded Systems (ICICES), International Conf* 1–9
23. Dukes L, Xiaohong Y, and Akowuah F 2013 *Proceedings of IEEE* 1–6
24. Shariar H and Zulkernine M 2011 *IEEE 9th International Conference on Dependable Autonomic and Secure Computing 2011* 7–14
25. Patil D, Patil K 2015 *International Journal of Computer Applications* **121(20)** 1–9