Information Security Controls against Cross-Site Request Forgery Attacks on Software Applications of Automated Systems

IOP Conf. Series: Journal of Physics: Conf. Series **1015** (2018) 042034 doi:10.1088/1742-6596/1015/4/042034

Information Security Controls against Cross-Site Request Forgery Attacks on Software Applications of Automated Systems

A V Barabanov[1], **A S Markov**[1], **V L Tsirlov**[2]

[1] Bauman Moscow State Technical University, 5, Baumanskaya 2-ya St., Moscow, 105005, Russia
[2] NPO Echelon, 24, Eletrozavodskaya St., Moscow, 107023, Russia

E-mail: A.Markov@bmstu.ru

Abstract. This paper presents statistical results and their consolidation, which were received in the study into security of various web-application against cross-site request forgery attacks. Some of the results were received in the study carried out within the framework of certification for compliance with information security requirements. The paper provides the results of consolidating information about the attack and protection measures, which are currently used by the developers of web-applications. It specifies results of the study, which demonstrate various distribution types: distribution of identified vulnerabilities as per the developer type (Russian and foreign), distribution of the security measures used in web-applications, distribution of the identified vulnerabilities as per the programming languages, data on the number of security measures that are used in the studied web-applications. The results of the study show that in most cases the developers of web-applications do not pay due attention to protection against cross-site request forgery attacks. The authors give recommendations to the developers that are planning to undergo a certification process for their software applications.

1. Introduction

Today, web-applications are actively used to build various automated control systems, including industrial ones. Despite the efforts of the developers, web-applications remain rather vulnerable. This is the underlying reason for improving methods of vulnerabilities identification in the specified software applications.

The methods for identifying vulnerabilities recommended by state controllers for information security (federal services) consist in the integrated use of approaches specified in the Common Criteria Evaluation Methodology (ISO/IEC 15408/18045) and ISO/IEC TR 20004 [1]. Unfortunately, more specific instructions for the testing laboratories have not been developed yet, which make the analysis of web-application vulnerabilities highly subjective.

The experience of evaluating web-application conformance shows that Cross-Site Request Forgery (CSRF-attack) is currently the most widely spread and successful attack. This can be explained by the fact that the developers of web-applications, as a rule, concentrate their attention on implementing measures protecting against attacks like SQL-injections or Cross-Site Scripting [2, 3].

It should be added that measures protecting against CSRF-attacks are still being actively studied, and best practices have not been clearly registered yet [4, 5].

The goal of this work consisted in developing guidelines for the developers of web-applications that are subject to conformance evaluation as per the information security requirements. To achieve the specified goal, this work (based on the experience of the certification tests) studied vulnerabilities of web-applications, CSRF-attacks and measures of protection against the specified class of attacks.

2. CSRF-Attack Concept

Vulnerabilities in web-applications related to incorrect implementation of the algorithm of HTTP-request authorization represent the main cause of a possibility of CSRF-attack implementation.

This usually happens if the following conditions are met [6, 7]:

- The browser automatically applies the user's authentication data (for instance, session cookie-files), when sending a HTTP-request to the web-application.
- The web-application uses obtained authentication data to authorize the actions required by the HTTP-request.

The following classification of CSRF-attacks can be offered (Fig. 1).

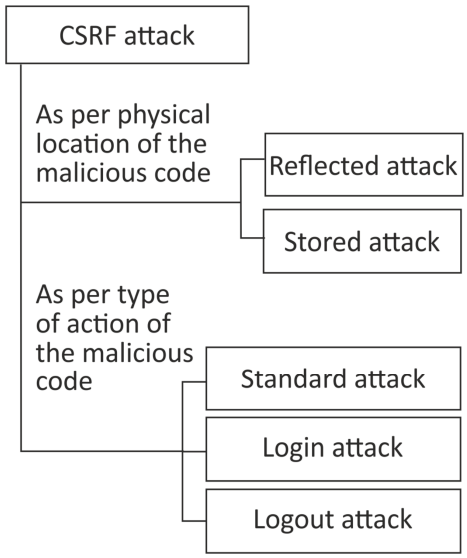

Figure 1. General Classification of CSRF-attacks.

It should be noted that despite difficulties in implementation, there are cases of successful CSRF-attacks of the 'Login' and 'Logout' type on web-applications [8]. The probability of successful 'stored' CSRF-attack is higher, because a malicious code is stored on the side of the attacked web-application, and the hacker does not have to make the user (for instance, using methods of social engineering) go to a special resource with a malicious code.

Implementation of the security measures on the client's side [9, 10, 11, 12, 13], represented by plugins/extensions of the browser or additional software (proxy), has significant drawbacks [8] and is currently only of academic interest. There are suggestions on implementing security measures using the browser software code directly, for instance, using 'samesite' properties of the cookie-files; however, currently these measures are experimental and are implemented only in certain browsers. Integrated measures (measures implemented jointly by the software code on the client- and the server-sides), as a rule, implement a certain information traffic control policy [7, 13], which contain critical information (for instance, authentication data), between the browser and the web-server. It should be noted that effective implementation of this type of security measures is possible by making changes in the browser software code. Moreover, essential limitations of these security measures are well-known, which does not allow their use as the sole measure of protection. The most popular security measures against CSRF-attacks are tokens (synchronic tokens or generated using HMAC cryptographic

functions) that are generated and tested on the web-application side. This security measure is implemented, as a rule, by the web-application itself or the framework.

The main distinctive feature of the token-based security measures is in the token storage method: it can be stored on the side of web-application and web-browser.

The leading specialists in the web-application security recommend using the echelon defence principle (multi-level protection), when implementing security measures. Thus, the specialists of OWASP community recommend implementing security of the web-application by combining two types of the security measures –HTTP-heading verification and tokens. In some cases, the developers use three or more security measures for critical information systems (for instance, bank servicing systems). For example, it can be a combination of tokens, verification of headings and security measures that require actions from the end user, who performs a critical operation (entry of one-time code/ password).

In view of the foregoing, it is possible to suggest the relevant classification of administrative and technical measures of protection against CSRF-attacks (Fig. 2).

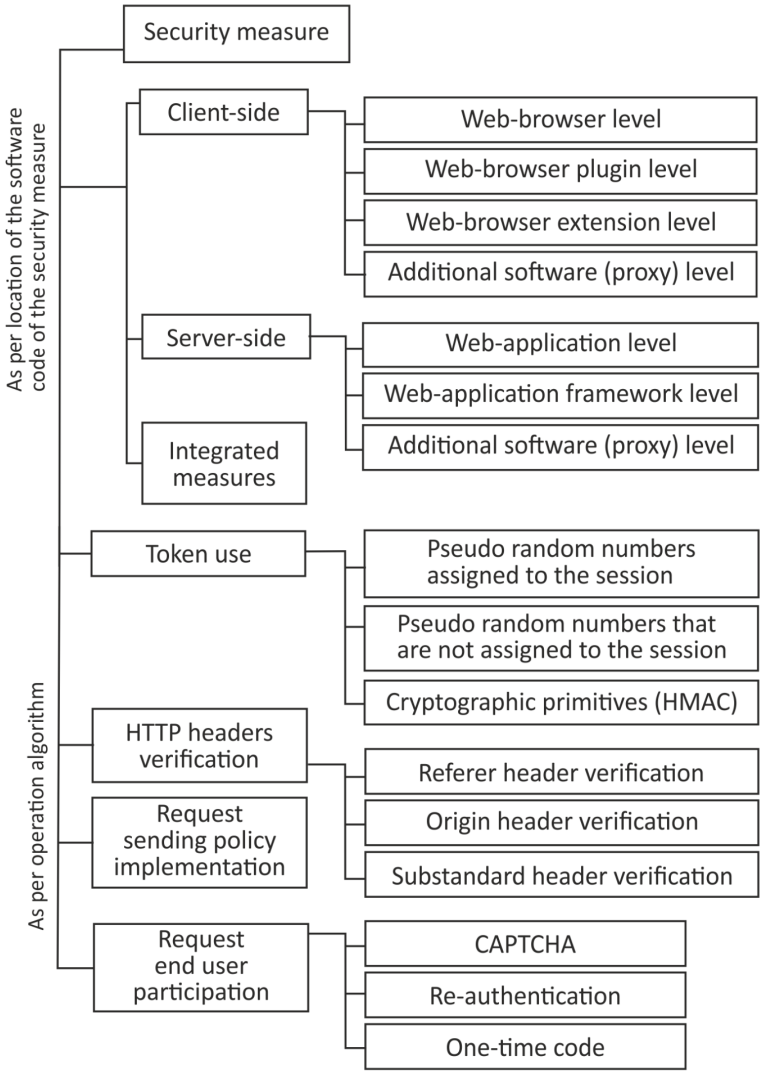

Figure 2. Classification of Security Measures against CSRF-attacks.

3. Methods of CSRF-Attack Study

The study into the security level of the web-application was carried out in the accredited test laboratory of NPO Echelon (study period: January – November 2017) [14].

Vulnerabilities were analysed using standard tests developed with account of recommendations [13 15] and CAPEC resource [16, 17]. Below is the general sequence of the performed tests:

1. Analysis of parts of web-applications (pages), which allow changing the condition of the web-application (creating/ changing/ deleting user accounts, protected information, other information etc.).
2. Study of the requests to the identified parts of web-applications: transmission of the requests from the web-browser to the web-application with further interception and analysis of the request structure. The expert analyses the intercepted request and defines the type of security measure against CSRF-attack on the specific page.
3. Generating a mock HTTP-request to be saved as an HTML-file on the local computer and is opened in the web-browser, provided that there is a session authenticated by the target of evaluation (web-application).
4. If the analysis of intercepted request (cl. 2) revealed security measures against CSRF-attacks, the following actions shall be additionally taken:
   a) When tokens are used as a security measure:
      - Analysis of URL for a clear token;
      - Sending a request without a token;
      - Sending a request with an altered token;
      - Sending a request using one token for various user accounts;
      - An attempt to guess /select a token;
   b) When using verification of the HTTP-package heading as a security measure:
      - Sending a request with altered HTTP Referer/Origin fields;
      - Sending a request without HTTP Referrer/Origin fields.

A standard test stand scheme used for the study can be seen in Figure 3.

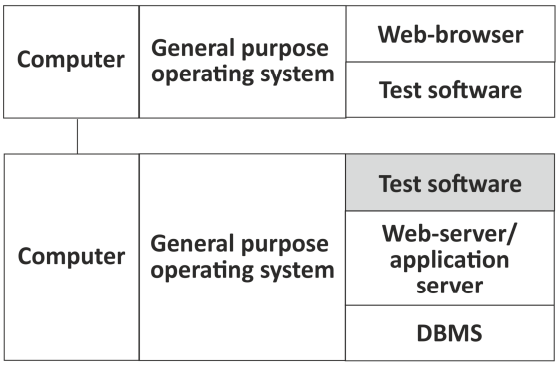

Figure 3. Test Stand Scheme.

The tests were performed using the following automation software: BurpSuite software, Scanner-VS software. The average time spent on testing of one web-application by one expert of the testing laboratory is 8 hours.

4. Study Results

The study yielded the following results:

- CSRF-attacks were successful in 70% of cases – 7 out of 10 analysed web-applications turned out to be vulnerable.
- The overwhelming majority of CSRF-attacks were successful in relation to web-applications developed in Russia (Figure 4). It should be noted that the only CSRF-attack that was

successful in relation to the foreign web-application was that of “Logout” type, and the experts of the testing laboratory failed to develop an attack vector that implements information security threat. Only one web-application initially did not have any security measures against CSRF-attacks. The other vulnerable web-applications had security measures based on verification of HTTP heading or token (Figure 5).

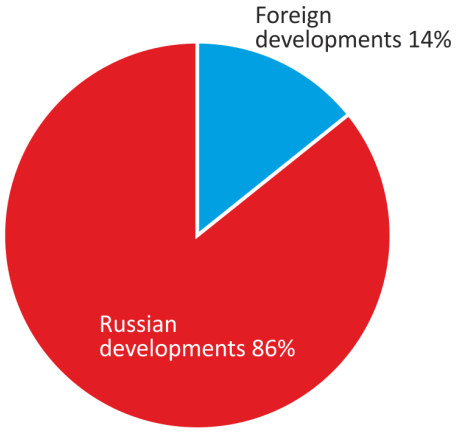

Figure 4. Distribution of the Identified Vulnerabilities as per the Developer Type.

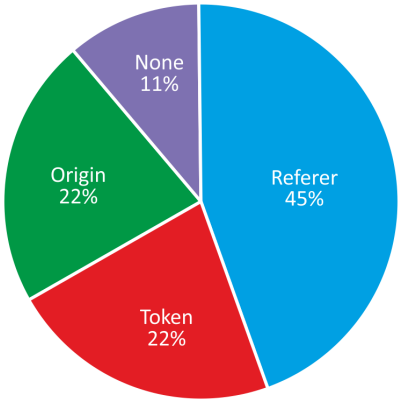

Figure 5. Distribution of Information Security Measures used in Vulnerable Web-Application.

- It has been established that web-applications written in PHP have a few more vulnerabilities that results in successful CSRF-attacks (Figure 6).

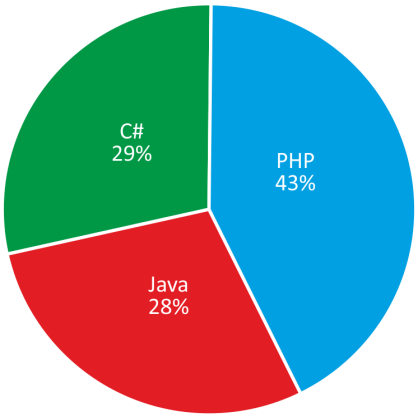

Figure 6. Distribution of identified vulnerabilities as to the programming language.

- The developers upgraded vulnerable web-applications using security measures based on tokens in all cases.
- In most cases, the upgraded web-application and web-applications, where the vulnerability has not been identified, used a combination of several security measures against CSRF-attacks (Figure 7).

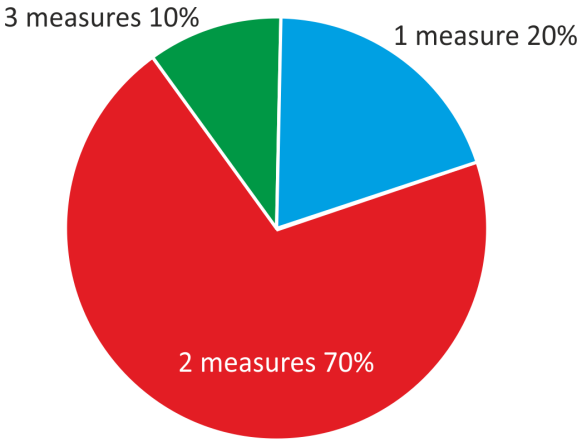

Figure 7. Data on the number of security measures used in the tested web-applications.

- The average time required for the web-application developer to correct a vulnerability is 3 weeks.

Thus the study showed that the majority of developers (around 70%) do not pay due attention to implementing security measures against the computer attacks under consideration.

5. Conclusions

The study allowed providing the following recommendations to the developers of web-applications of the automated systems, which fall under the requirements of the technical regulations (conformance evaluation).

- It is advisable that the developers implement measures of secure software development in the software lifecycle processes. The priority shall be given to implementing measures related to testing penetration of web-applications. However, it is important that the standard tests are supplemented with the test for exposure to CSRF-attacks of “Login” and “Logout” type.

- The developers are advised to use the echelon defense principle (security in depth), which consists in combining two or more security measures (for instance, verification of token and HTTP-heading), when implementing security measures against CSRF-attacks in the web-application.
- When implementing security measures against CSRF-attacks in the web-application, the developers are, first of all, recommended to use security measures that have already been implemented in the operational environment, for instance, in frameworks.

References

[1] Barabanov A and Markov A 2015 Modern trends in the regulatory framework of the information security compliance assessment in Russia based on common criteria *The 8th international conference on security of information and networks (SIN '15)* 30-33
[2] Babincev I M and Vuletic D V 2016 Web application security analysis using the kali linux operating system *Vojnotehnicki glasnik* **64 2** 513-531
[3] Reber G, Malmquist K and Shcherbakov A 2014 Mapping the application security terrain *Voprosy kiberbezopasnosti* **1(2)** 36-39
[4] Jayaraman W Du K, Tan X, Luo T and Chapin S 2011 *The 2011 new security paradigms workshop (NSPW '11)* 83-94
[5] Jovanovic N, Kirda E and Kruegel C 2006 *The ieee international conference on security and privacy for emerging areas in communication networks (SecureComm 2006)* 1–10
[6] Czeskis A, Moshchuk A, Kohno T and Wang H J 2013 *The 22nd international conference on world wide web (WWW '13)* 273-284
[7] Jayaraman K, Talaga P G, Lewandowski G, Chapin S J and Hafiz M 2009 *The 16th conference on pattern languages of programs (PLoP '09)* 16 1-9
[8] Barth A, Jackson C and Mitchell J C 2008 *The 15th acm conference on computer and communications security (CCS '08)* 75-88
[9] Shahriar H and Zulkernine M 2010 *The 2010 ieee 21st international symposium on software reliability engineering (ISSRE '10)* 358-367
[10] Ryck P D, Desmet L, Heyman T, Piessens F and Joosen W 2010 *The second international conference on engineering secure software and systems (ESSoS'10)* 18-34
[11] Pelizzi R and Sekar R 2011 *The 27th annual computer security applications conference (ACSAC '11)* 257-266
[12] Xing L, Zhang Y and Chen S 2010 *The 13th international conference on recent advances in intrusion detection (RAID'10)* 484-485
[13] Maes W, Heyman T, Desmet L and Joosen W 2009 *The first acm workshop on secure execution of untrusted code (SecuCode '09)* 3-10
[14] Barabanov A V, Lavrov A I, Markov A S, Polotnyanschikov I A and Tsirlov V L 2017 The study into cross-site request forgery attacks within the framework of analysis of software vulnerabilities Trudy *ISP RAN/Proc. ISP RAS* **29 5** 7-18
[15] Barabanov A V, Markov A S and Tsirlov V L 2016 Methodological framework for analysis and synthesis of a set of secure software development controls *Journal of Theoretical and Applied Information Technology* **88 1** 77-88
[16] Gelernter N and Herzberg A 2015 *The 22nd acm sigsac conference on computer and communications security (CCS '15)* 1394-1405
[17] Li X and Xue Y 2014 *ACM Comput. Surv.* **46 4** 29