Scientific and Practical Cyber Security Journal (SPCSJ) 2(1):22-28
Scientific Cyber Security Association (SCSA), 2018 ISSN: 2587-4667

THE MATHEMATICAL MODEL OF THE TESTING TECHNOLOGY FOR DOM XSS VULNERABILITIES

O.Kovalenko, A.Kovalenko, O.Smirnov, S.Smirnov, V.Vialkova
Kirovohrad National Technical University
Taras Shevchenko National University of Kyiv

ABSTRACT: The paper presents the results of the study and a mathematical model of the testing technology for vulnerability to one of the most common types of attacks on Web applications – XSS (Cross Site Scripting) – XSS DOM. Cross-site scripting is an error validation of user data, which allows you to send JavaScript code to be executed in the user's browser. Attacks of this kind are often also referred to as HTML injection, because the mechanism of their implementation is very similar to SQL injection, but unlike the latter, the introduced code is executed in the user's browser. The approach of mathematical modeling based on GERT networks is argued. Studies have shown that GERT (Graphical Evaluation and Review Technique) is a method of studying and analysis of stochastic networks that are used to describe the logical relationship between parts of the project or stages of the process. The main purpose of GERT is to evaluate the logic of the network and the duration of activity and reception of the conclusion about necessity of execution of some activities. A mathematical model of the testing technology of Web applications is developed. As the basis of the mathematical modeling the approach of GERT-network synthesis was taken. The developed mathematical model of testing technologies of DOM XSS vulnerability differs from the known by accounting of performance or analyzing the DOM structure. The developed mathematical model can be used when testing the vulnerability of a Web application.

KEYWORDS: testing, DOM XSS vulnerabilities, GERT-network, security vulnerabilities.

1. Formulation of the problem

Currently a large demand for Web applications and Web services causes a great interest of the attackers to their possible vulnerabilities. However, the main threats towards the server component were transformed into attacks against ordinary users.

The analysis of materials of the Open Web Application Security Project (OWASP TOP 10) have shown that one of the most dangerous types of attacks (vulnerability) is a cross-site scripting – XSS (Cross Site Scripting).

Analysis of the literature showed that cross-site scripting is an error validation of user data, which allows you to send JavaScript code to be executed in the user's browser. Attacks of this kind are often also referred to as HTML injection, because the mechanism of their implementation is very similar to SQL injection, but unlike the latter, the introduced code is executed in the user's browser.

From [1-8] it is known that under the XSS is usually meant instant and delayed cross-site scripting. At the instant malicious XSS code (Javascript) is returned by the target server immediately in response to a HTTP request. Deferred XSS means that malicious code is stored on the target system and can be later embedded into an HTML page on an affected system. This classification assumes that a fundamental property of XSS is that malicious code is sent from the browser to the server and returns to the same browser (snapshot XSS) or any other browser (delayed XSS).

The big number of Internet articles describe in detail the basic mechanisms of the emergence of such threats and the possible ways of blocking. However, to identify these threats and the possible consequences of their distribution in the process of security management of IT projects and to offer the best solutions to this problem, there is a need of a mathematical formalization of the process of initiation and propagation.

Especially urgent task in this direction is the modeling of the DOM (Document Object Model) XSS vulnerability. This is due to the fact that the DOM XSS vulnerability is a subspecies of XSS, where the attack is not in the server's response and, accordingly, not in the HTML code and the DOM structure of the HTML page. The results of attacks through these vulnerabilities can only be detected during execution or analysis of the DOM structure. The mechanism of attack, namely the injection of Javascript code in the affected segment remains unchanged.

The aim of this work is to develop mathematical models of the technology of testing for vulnerability to one of the most common types of attacks on Web applications – DOM XSS.

2. Statement of the main material

The algorithm for the DOM XSS vulnerability analysis

For mathematical formalization of the algorithm that parses the DOM XSS vulnerabilities we use the basic provisions of the GERT network modeling, described in detail [9-11].

The algorithm for the analysis of DOM XSS vulnerability can be described as follows:

1) From the code of the analyzed page all the <script> tags are extracted and a list of tags for analysis is compiled.

2) The analysis of contents of the tag is parsed. If the tags do not contain code, and refer to the deleted file, acces the file and get the code fromit. The contents of the file are potential unsafe sections of code (the sink) that use client input (source).

Examples of sources include:

- document.URL
- document.documentURI
- location.href
- location.search
- location.*
- window.name
- document.referrer

Examples of sink:

- document.write
- (element).innerHTML
- eval
- setTimout / setInterval
- execScript

3) If the code tag is used by source, attack with a specific marker that can be tracked in the DOM structure of the page is running after executing the code (e.g., injection of a specific text content in the DOM).

4) The content DOM is checked. If the attack marker is in the DOM, we can conclude about the presence of DOM vulnerability.

5) Steps 2 – 4 are executed for each script tag on the page.

In order to build a formal model of the algorithm of vulnerability analysis of Web applications to a DOM XSS stochastic GERT - network is selected.

Studies have shown that GERT (Graphical Evaluation and Review Technique) is a method of studying and analysis of stochastic networks that are used to describe the logical relationship between parts of the project or stages of the process [9-11]. The main purpose of GERT is to evaluate the logic of the network and the duration of activity and reception of the conclusion about necessity of execution of some activities.

GERT network consists of nodes of type AND, INCLUSIVE-OR and EXCLUSIVE-OR, and branches with two or more parameters. The branch has a direction, has the node start and node end. The parameters of the branches contain:

1) the probability of the branches (Pa) under the condition that the node that is the source of the branches has been implemented;

2) the time (ta) passing branches, if it will be implemented.

The time ta may be a random value. If the branch is not part of the network implementation, i.e. during the execution of the process activity associated with the branch does not occur, then ta = 0.

The node in the stochastic network GERT consists of a input function (contributive functions) and output functions (distribution functions). Each function describes a specific logical relation relative to the associated branches.

In General, studies have shown that GERT-modeling is an effective way of identifying previously unknown laws and distribution functions of random variables with known algorithm of functioning (process). Therefore, as a tool of mathematical modeling we have chosen the GERT modeling.

GERT-model of the mathematical model of the testing technology for DOM XSS vulnerabilities

Let's construct, in accordance with the description of the GERT network, a model of the mathematical model testing technologies of DOM XSS vulnerability. A graphical image of the GERT model is shown in Fig. 1.

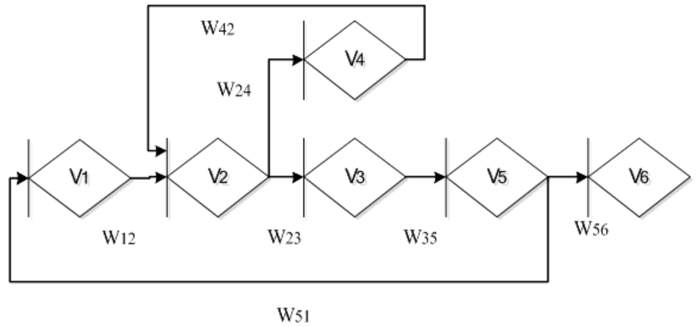

***Figure 1**. GERT-model of the mathematical model of testing technology for DOM XSS*

In the presented network the nodes of the graph are interpreted by the States of computer system in the operation of the DOM structure, and the branches of the graph and probabilistic-temporal characteristics of state transitions. In particular, branch (1,2) describes the collection and analysis of tag content. Branch (2,3) shows the time characteristics of execution of the attack in the case of " source" structure. Branch (2,4) specifies the random access to the contents of the deleted file (search for "sink"). Branch (4,2) characterizes the return on the execution of the attack. Branch (3,5) describes the continuation of the attack, in particular checking the contents of the

DOM. Next branch (5,6) characterizes the time of the decision about the vulnerability, at the same time branch (5,1) displays the temporal characteristics of the transition to the new tag.

Characteristics of the branches of the model are presented in table. 1

Table 1. *Characteristics of the branches of the model*

№	Branch	W-function	Probability	The generating function of the moments
1	(1,2)	W_{12}	p_1	$\lambda_1/(\lambda_1-s)$
2	(2,3)	W_{23}	p_2	$\lambda_2/(\lambda_2-s)$
3	(2,4)	W_{24}	p_3	$\lambda_3/(\lambda_3-s)$
4	(3,5)	W_{35}	p_2	$\lambda_2/(\lambda_2-s)$
5	(5,6)	W_{56}	p_4	$\lambda_4/(\lambda_4-s)$
6	(5,1)	W_{51}	$1-p_4$	$\lambda_5/(\lambda_5-s)$
7	(4,2)	W_{42}	p_3	$\lambda_3/(\lambda_3-s)$

Equivalent W-function of execution time of mathematical model testing technologies of DOM XSS vulnerability is equal to:

$$W_E(s)=\frac{W_{12}W_{23}W_{35}W_{56}+W_{12}W_{24}W_{42}W_{23}W_{35}W_{56}}{1-W_{12}W_{23}W_{35}W_{51}-W_{12}W_{24}W_{42}W_{23}W_{35}W_{51}}=$$

$$=\frac{p_1p_2^2\lambda_1\lambda_2^2\left(p_4\lambda_4(\lambda_3-s)^2(\lambda_5-s)+p_3^2q_1\lambda_3^2\lambda_5(\lambda_4-s)\right)}{(\lambda_4-s)\begin{pmatrix}(\lambda_1-s)(\lambda_2-s)^2(\lambda_3-s)^2(\lambda_5-s)- \\ -p_1\lambda_1p_2^2\lambda_2^2q_1\lambda_5(\lambda_3-s)^2-p_1p_2^2p_3^2\lambda_1\lambda_2^2q_1\lambda_3^2\lambda_5\end{pmatrix}}, \qquad (1)$$

where $1-p_4=q_1$.

The peculiarity of the process lies in the heterogeneity of the analyzed and processed data. There can be multiple instances of the feedback. In Fig. 1 these cycles are recorded in the form of transitions $W_{12}\to W_{24}\to W_{42}$ $W_{12}\to W_{23}\to W_{35}\to W_{51}$

For GERT-networks with cycles, there are no simple methods of finding singular points of the function $\Phi_E(z)$ replacing the real variables $(z=-i\varsigma)$, where ς the variable is valid. This is because of the fact that for finding the singular points it is necessary to solve nonlinear equations, and the more complex the structure of the GERT-network, the more difficult and the original equation. Therefore, during the simulation it is proposed to resort to such replacement.

Completing a comprehensive transformation $z=-s$, we get

$$\Phi(z)=\frac{uz^3+vz^2+bz+k}{(\lambda_4+z)\left(z^6+cz^5+dz^4+gz^3+hz^2+wz+m\right)}, \qquad (2)$$

where:

$u=-p_1p_2^2p_4\lambda_1\lambda_2^2\lambda_4$,

$v=p_1p_2^2p_4\lambda_1\lambda_2^2\lambda_4(\lambda_5+2\lambda_3)$,

$b=-p_1p_2^2p_4\lambda_1\lambda_2^2\lambda_4\lambda_3(2\lambda_5-\lambda_3)$,

$k=-p_1p_2^2\lambda_1\lambda_2^2\lambda_3^2\lambda_4\lambda_5\left(p_4+p_3^2q_1\right)$,

$c=\lambda_1+2\lambda_2+2\lambda_3+\lambda_5$,

$d=-(2\lambda_3\lambda_5+\lambda_1\lambda_5+2\lambda_2\lambda_5+\lambda_3^2+2\lambda_1\lambda_3+4\lambda_2\lambda_3+2\lambda_1\lambda_2+\lambda_2^2)$,

$g=\lambda_3^2\lambda_5+4\lambda_1\lambda_2\lambda_5+4\lambda_2\lambda_3\lambda_5+\lambda_2^2+\lambda_3^2\lambda_1+2\lambda_3^2\lambda_2+4\lambda_1\lambda_2\lambda_3+2\lambda_2^2\lambda_3+\lambda_2^2\lambda_1$,

$h=-(\lambda_1\lambda_3^2\lambda_5+2\lambda_2\lambda_3^2\lambda_5+4\lambda_1\lambda_2\lambda_3\lambda_5+2\lambda_2^2\lambda_3\lambda_5+\lambda_2^2\lambda_3^2+2\lambda_1\lambda_2^2\lambda_3-p_1p_2^2q_1\lambda_1\lambda_2\lambda_5)$,

$w=\lambda_1\lambda_2\lambda_3^2\lambda_5+\lambda_2^2\lambda_3^2\lambda_5+2\lambda_1\lambda_2^2\lambda_3\lambda_5+\lambda_1\lambda_2^2\lambda_3-2p_1p_2^2q_1\lambda_1\lambda_2\lambda_3\lambda_5$,

$m=p_1p_2^2q_1\lambda_1\lambda_2\lambda_3^2\lambda_5+p_1p_2^2p_3q_1\lambda_1\lambda_2^2\lambda_3^2\lambda_5-\lambda_1\lambda_2^2\lambda_3\lambda_5$.

The probability density function of the execution time of algorithm analysis of DOM XSS vulnerability:

$$\varphi(x)=\frac{1}{2\pi i}\int_{-i\infty}^{i\infty}e^{zx}\frac{uz^3+vz^2+bz+k}{(\lambda_4+z)\begin{pmatrix}z^6+cz^5+dz^4+\\+gz^3+hz^2+wz+m\end{pmatrix}}dz, \qquad (3)$$

where the operation of integration is performed using the integral of Bromwich-Wagner [11]

Integration method depends on whether the function $\Phi(z)$ has only simple poles or poles of some order. In the case when the function $\Phi(z)$ has only simple poles, the expression $å^{zx}\Phi(z)$ can be represented as:

$$e^{zx}\Phi(z)=\frac{e^{zx}\left(uz^3+vz^2+bz+k\right)}{z^7+\gamma_6z^6+\gamma_5z^5+\gamma_4z^4+\gamma_3z^3+\gamma_2z^2+\gamma_1z+\gamma_0}=\frac{\mu(z)}{\psi(z)}, \qquad (4)$$

где: $\gamma_6=\lambda_4+c$, $\gamma_5=c\lambda_4+d$, $\gamma_4=d\lambda_4+g$, $\gamma_3=g\lambda_4+h$, $\gamma_2=h\lambda_4+w$, $\gamma_1=w\lambda_4+m$, $\gamma_0=m\lambda_4$.

Then the density distribution of the execution time of the algorithm of analysis of DOM XSS vulnerability is equal to:

$$\varphi(x)=\sum_{k=1}^{7}\mathrm{Res}\left[e^{zx}\Phi(z)\right]=\sum_{k=1}^{7}\frac{\mu(z_k)}{\psi(z_k)}=\sum_{k=1}^{7}\frac{e^{zx}\left(uz^3+vz^2+bz+k\right)}{7z_k^6+6\gamma_6z_k^5+5\gamma_5z_k^4+4\gamma_4z_k^3+3\gamma_3z_k^2+2\gamma_2z_k+\gamma_1} \qquad (5)$$

Function $\Phi(z)$ except for decisions determined by the roots of the equation $z^6+cz^5+dz^4+gz^3+hz^2+wz+m=0$, can have a pole of second or third order cases where the value λ_4 is equal to the value of the roots z_2, z_3, z_4, z_5, z_6, z_7. In these cases, the density of distribution of time of message transmission $\varphi(x)$ is in the formula for finding deductions r_{-1} from the poles z_k with n order:

$$r_{-1}=\frac{1}{(n-1)!}\lim_{z\to z_k}\frac{d^{n-1}\left((z-z_k)^n e^{zx}\Phi(z)\right)}{dz^{n-1}}. \qquad (6)$$

The expression (6) is a rational function z with relative degree of denominator larger than degree of numerator. Therefore, it is performed the conditions of the Lemma of Jordan [11]. The function $\Phi(z)$has poles at the points $z_1=-\lambda_4$. The polynomial: $z^6+cz^5+dz^4+gz^3+hz^2+wz+m$ generates seven poles. The solution to the equation

$$z^6+cz^5+dz^4+gz^3+hz^2+wz+m=0 \qquad (7)$$

can be found by any method, for example, by the formulas of Viett [11]. In result singular points z_2, z_3, z_4, z_5, z_6, z_7 are computed.

Thus, based on the exponential GERT-network developed mathematical model of the algorithm of analysis of DOM XSS vulnerability that differs from the known, subject to the execution or analysis of the DOM structure.

The model can be used to study processes in computerized systems, the development of new tools and data security protocols.

The use of exponential stochastic models GERT will give an opportunity to use the results obtained in the analytical form (functions, density distribution) for comparative studies and research, and more complex computer systems mathematical methods.

3. Conclusions

In this work the mathematical model of the technology of testing WEB applications is developed. The basis of the mathematical modeling is connected to the approach of GERT-network synthesis. In result, the mathematical model testing technologies a DOM XSS vulnerability is developed.

Mathematical model of testing technologies of DOM XSS vulnerability differs from the known, subject to the execution or analysis of the DOM structure that makes it possible to conduct an analytical evaluation of the time spent testing this vulnerability in the context of realization of strategy of development of secure software.

The while researching, it was determined that a random variable execution time of the considered test methods in general corresponds to the gamma distribution. Verification of this hypothesis is produced by the χ^2 Pearson criterion.

REFERENCES

1. About The Open Web Application Security Project – OWASP: https://www.owasp.org/index.php/About_The_Open_Web_Application_Security_Project.
2. OWASP Top 10 – 2017 RC1: https://github.com/OWASP/Top10/blob/master/2017/OWASP%20Top%2010%20-%202017%20RC1-English.pdf.
3. Positive Research 2016: https://www.ptsecurity.com/upload/ptru/analytics/Positive-Research-2016-rus.pdf.
4. OSSTMM 3 – The Open Source Security Testing Methodology Manual. Contemporary Security Testing And Analysis: http://www.isecom.org/mirror/OSSTMM.3.pdf.
5. Testing for DOM-based Cross-site scripting (OTG-CLIENT-001) – OWASP: https://www.owasp.org/index.php/Testing_for_DOM-based_Cross_site_scripting_(OTG-CLIENT-001).
6. Testing for SQL Injection (OTG-INPVAL-005) – OWASP: https://www.owasp.org/index.php/103 Testing_for_SQL_Injection_(OTG-INPVAL-005).
7. Cohen W., Ravikumar P., Fienberg S. A Comparison of String Metrics for Matching Names and Records / William W. Cohen, Pradeep Ravikumar, Stephen E. Fienberg.: https://www.cs.cmu.edu/afs/cs/Web/People/wcohen/postscript/kdd-2003-match-ws.pdf.
8. Kevin Dreßler a , Axel-Cyrille Ngonga Ngomo On the Efficient Execution of Bounded Jaro-Winkler Distances / Semantic Web – Interoperability, Usability, Applicability an IOS Press Journal http://www.semantic-web-journal.net/system/files/swj944.pdf
9. Pritsker A. A. B. GERT: Graphical Evaluation and Review Technique. Part I. Fundamentals / Pritsker A. A. B., Happ W. W. // The Journal of Industrial Engineering (May 1966). pp. 267-274.
10. Pritsker, A. A. B. Modeling and analysis using Q-GERT networks / Pritsker, A. A. B. – New York: Wiley : Distributed by Halsted Press, 1979 – 435 p.

11. Semenov S.G., Zmiyevskaya V N., Kassem Khalife Development of Gert model of management system by using test cases // Journal of Qafqaz university-mathematics and computer science 2016, Vol.(4), № 1 PP. 52-59.