Research on Client-side Defense Techniques of Cross-Site Scripting Attack

Xuyang Wang and Mingyang Xu
School of Computer & Communication, Lanzhou University of Technology, Lanzhou730050, Lanzhou, China

Keywords: Cross-site scripting; Browser security; Dynamic data tainting; Static data tainting; JavaScript engine

Abstract. The Cross-site scripting (XSS) is among the most serious and common threat in Web application today. The main purpose of XSS is to steal the user's sensitive information, as its behavior is to send user's sensitive information to a third party without the user's authorization,we can get the XSS attack detection results by analyzing the situation of user's accessing sensitive information in current page. The detection technique presented in this paper adopts the idea of protecting user information in client-side of the Web browser. By analyzing its JavaScript engine, we extend its handle process in each phase. Our approach employs dynamic analysis techniques in general, and an auxiliary static analysis technique when necessary to analyze the situation of sensitive information in current page. By handling and judging the analysis result, we can prevent the suspicious XSS attack. If sensitive information is about to transferred to a third party, the user can decide id this should be permitted or not. The result of our experiment has demonstrated that the behavior-based XSS detection technique proposed in this paper is feasible in practice model.

Introduction

Web 2.0 applications are very attractive to the developers and end-users [1], because they provide friendly interface, plenty of functions and high practicality. The main purpose of XSS is to steal the user's sensitive information, as its behavior is to send user's sensitive information to a third party without the user's authorization, we can get the XSS attack detection results by analyzing the situation of user's accessing sensitive information in current page [2]. Proposed method of cross-site script defense is a new pure client-side XSS defense method. You can form a client to deal with cross-site attack an effective barrier, enhance the client's active defense capabilities. It focuses on dynamic taint tracking, supplemented by static stain tracking. Which combines the high efficiency of dynamic labeling and the accuracy of static analysis, the overall framework of the defense method is shown in Fig. 1-1.

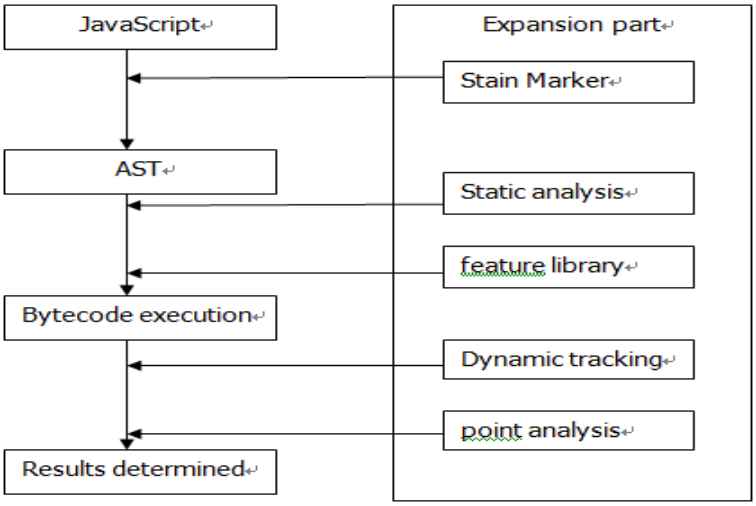

Figure 1. Framework of defense

Summarized is as follows:

(1) This technique first marks the sensitive data contained in the web page that the user is currently accessing;

(2) To determine whether the stain data exists in the database known feature library, If present, a warning is issued, Submitted to the user for processing;

(3) If it does not exist,The process used in the taint data browser is followed in real time. Once the stain data is found to be sent to a third party, To immediately take a variety of measures, Such as logging, Prevents data transfer, Or terminate the program and pop up the error, Submitted to the user processing;

(4) For newly discovered stains, Store them in a known feature library, This greatly saves the detection of the same stain after the time.

Tracking of Dynamic Stain

Dynamic stain tracking is the core idea, The user to visit the current page sensitive data in the stain mark, In principle, tainted data can only be sent to its source site [3]. The operation of the stain data in the browser is then followed, once the stain data is suspected of operation, tainted data can be exploited by attackers, Tracking system will immediately send an alarm signal, It is up to the user to decide whether or not to allow the operation[4].

Selection of Sensitive Information Source. In order to realize the dynamic tracking of stain information, The taint information must be marked before the client script code is executed, What type of information should be classified as tainted information is the first consideration, To this end, the concept of information sources is introduced[5].

Table 2-1 Initial sources of taint values

object	Attributes of mark
Document	Cookie, domain, forms, last Modified, links, referrer, title, URL
Form	Action
Any form input element	Checked, default Checked, default Value, name, selected Index, to String, value
History	Current, next, previous
Select option	Default Selected, selected, text, value
Location and Link	Hash, host, hostname, pathname, port, protocol, search
Window	Default Status, status

When a data source contains information that would be exploited by an attacker to launch an attack, or when the user information is obtained[6], this data source is defined as a sensitive information source. The paper uses the source information provided by Netscape. Shown in Table 2-1, the form is relatively comprehensive, when a new sensitive information source is discovered, the system also facilitate to extend the expansion of the form.

Analysis of Stain Transmission

JavaScript program as a part of the web page, which is first parsed in the client browser, is compiled into internal byte codes and followed by the JavaScript engine processing[7]. In order to track the use of sensitive information in JavaScript program, the function of JavaScript engine is need to

extend, which is embodied in the semantics of extended byte codes instruction, so that the stain information can be effectively diffused. JavaScript byte codes instruction can be divided into the following four operations: Assignment operation, Arithmetic and logic operation (+,-), Controlling and looping statement(if, while , switch), the function call and eval.

When an instruction is executed, its operands may be tainted data. Therefore, it is necessary to define rules that specify whether the instruction result should be marked as a taint data for different operating instructions. Effective delivery and diffusion of stain data is ensured.

(1) Assignment operation

The assignment operation is a constant or an expression value is assigned to a left-value (the left value of equation, the left value is equal to variable numerical). If the constant or the value of the expression (i.e. a right-value) has been marked as tainted data, then after the end of operation a left-value will be marked as tainted data.

(2) Arithmetic and logic operation

Where a certain operand is marked with a smudge, the result of the operation is marked as a tainted data.

(3) Controlling and looping statement

In the if-else statement, if the statement needs to determine the information that is marked by the taint, and the scope will cover both the if-block and the else-block. In the try-catch-finally block, the taint scope will overwrite the catch-block after the detected data in the try-block is detected with tainted data. In the Do.. While statement, the Do body is for the first time executed , and it does not have any stain scope, only when the implementation of the while instruction, and while conditions in the tainted data is detected, the scope will be stain covering the entire body of the loop, when the cycle ends, the scope is canceled[8].

(4) Controlling and looping statement

Eval (String) is the basic syntax of the eval function, the JavaScript expression is needed to calculate, or the execution statement contains in the string parameters. When the string parameter is marked as tainted data, or called in the tainted data scope to execute the eval function code, the procedures for the implementation of Eval function block would be tracing scope for execution in stain, the operation will be tracking.

Tracking of Static Stain

From the above analysis, the dynamic stain tracking can effectively monitor data dependence and direct control dependencies, thus it can efficiently track sensitive information flow [9]. However, dynamic tracing technology can not monitor indirect control dependencies. As a supplement to dynamic tracking, Static analysis can solve this problem, when dynamic tracking can not effectively monitor the flow of information has the risk of leakage of sensitive information, the static analysis method is invoked.

In this paper, the constraint analysis method is applied, based on the information flow model, design and implementation of the JavaScript script constraint system. Through the constraint analysis, to achieve stain diffusion, the following is a step-by-step introduction to static taint tracking.

Constraint Analysis System. The constraint analysis system is based on the information flow model, information flow model, the data are kept independent of each other、On a delimited interval defined by the security level[10]. The Bell-Lapadula model is an information flow model used to provide confidentiality, concerned about the different levels of security data flow between the problem. Learning from Bell-Lapadula and Bida's classic security model, the following rules are established in this paper.

CONFIDENTIAL F-NR: The stain information is not leaked $(x,\square)$

Integrity F-NW: The taint information can not be tampered $(\square, x)$

Assignment statement

The constraint analysis rules for assignment statements are shown in Equation 3-1, first constrain the expression e, Subsequent analysis the direct information of e flow to x, and the indirect information flow associated with x, Similarly, Equation 3-2 is the constraint analysis of the return statement.

$$Con(k, x = e, x_i) = Con(k, e, x_i) \bigcup \{x_e \subset x_x\} \bigcup \{x_i \subset x_x\} \tag{3-1}$$

$$Con(k, returne, x_i) = Con(k, e, x_i) \bigcup \{x_e \subset x_{ret_k}\} \bigcup \{x_i \subset x_{ret_k}\} \tag{3-2}$$

(1) Control statement

When multiple statements are executed in sequence, each constraint is analyzed in turn, As shown in Equation 3-3. Conditional control statements, loop control statements of the constraint analysis rules are 3-4and 3-5.

$$Con(k, s_1; s_2, x_i) = Con(k, s_1, x_i) \bigcup Con(k, s_2, x_i) \tag{3-3}$$

$$Con(k, if_j xthens_1 esles_2, x) = Con(k, s_1, x_i) \bigcup Con(k, s_2, x_i) \bigcup \{x_i \subset x_j\} \bigcup \{x_x \subseteq x_j\} \tag{3-4}$$

$$Con(k, while_l xdos, x_i) = Con(k, s, x_i) \bigcup \{x_i \subseteq x_l\} \bigcup \{x_x \subseteq x_l\} \tag{3-5}$$

(2) Definition and call of function

Constraint Analysis of Function Calls As shown in Equation 3-6, the this pointer points to the context object. Similarly, the method call will be regarded as special attributes, as shown in Figure 3-7.

$$Con(k, f(e')ase, x_i) = Con(k, e', x_i) \bigcup \left\{x_f \subseteq Fun(\phi, x_{obj_{context}}, x_{e'}, x_e, x_i)\right\} \tag{3-6}$$

$$Con(k, f(e')ase, x_i) = Con(k, e', x_i) \bigcup \left\{x_x \subseteq \mathrm{Re}al\left(Attr_f(\phi, Fun(\phi, x_x, x_{e'}, x_e, x_i))\right)\right\} \bigcup \left\{x_x \subseteq \mathrm{Pr}o\left(Attr_f(\phi, Fun(\phi, x_x, x_{e'}, x_e, x_i))\right)\right\} \tag{3-7}$$

(3) Object properties

The operation of the object attribute is divided into reading and assignment, and its constraint analysis rules are shown in Eqs. 3-8 and 3-9, respectively. Can read the property of the inherited object and the direct object information flow to the read attribute, you can modify the attribute inheritance object and the direct object information flow assignment attribute.

$$Con(k, x.fase, x_i) = \left\{x_x \subseteq \mathrm{Re}al\left(Attr(\phi, x_e)\right)\right\} \bigcup \left\{x_x \subseteq \mathrm{Pr}o\left(Attr_f(\phi, x_e)\right)\right\} \tag{3-8}$$

$$Con(k, x.f = e, x_i) = Con(k, e, x_i) \bigcup \left\{x_x \subseteq \mathrm{Re}al\left(Attr_f(x_e, \phi)\right)\right\} \bigcup \left\{x_x \subseteq \mathrm{Re}al\left(Attr_f(x_i, \phi)\right)\right\} \tag{3-9}$$

Constraint Solving

The detailed rules for the establishment of the constraint system are outlined in detail in the previous section. In this section, $x \subseteq F - NR$ the stain rule is used to solve the problem. In the case of $x_1 \to x_n$ the stain information, $x_1 \to x_n$ we call NR (x) to obtain the relevant information

flow from x, $x_i \subseteq F - NR$ and $x \subseteq F - NW$ the information nodes are marked as smear information, ie, $x_i \subseteq F - NW$ the same, if so, to achieve the diffusion of stain.

Analysis of Transmission Point and to Deal Sensitive Information

Through the above dynamic and static stain tracking, this article's cross-site scripting attack defense system completed the spread of the stain, the formation of effective tracking of the stain information[11]. But only to the sensitive information collected to the attacker to control the site, that the stain data is transferred to a third party, a cross-site scripting attack can be successful. There are several ways to do this:

1) Modify the source address of pictures in the page;
2) Set location of document to change the current web page address;
3) Automatically submit web form;

If you encounter the above transfer operation, it is first determined whether or not the data contents of the transmission operation contain the stain information, if present, then it is analyzed whether the destination of the transmission operation conforms to the homology policy, if not meet the suspicious cross-site attacks, The user is warned, It is up to the user to decide whether to allow the transfer operation.

Conclusions

Web 2.0 applications are very attractive to the developers and end-users, because they provide friendly interface, plenty of functions and high practicality. The main purpose of XSS is to steal the user's sensitive information, as its behavior is to send user's sensitive information to a third party without the user's authorization. In the implementation, this paper chooses the open-source Web browser Mozilla Firefox as its experimental platform. By analyzing its JavaScript engine, we extend its handle process in each phase. Our approach employs dynamic analysis techniques in general, and an auxiliary static analysis technique when necessary to analyze the situation of sensitive information in current page. By handling and judging the analysis result, we can prevent the suspicious XSS attack. If sensitive information is about to transferred to a third party, the user can decide id this should be permitted or not. The result of our experiment has demonstrated that the behavior-based XSS detection technique proposed in this paper is feasible in practice.

Reference

[1] KHIN SHAR L, Kuan Tan H B. Defending against cross-site scripting stacks [J].Computer, 2012, 45(3): 55-62.

[2] Van Gundy M, Chen H. Noncespaces: Using randomization to defeat cross-site scripting attacks [J]. Computers & Security, 2012, 31(4): 612-628.

[3] Dafydd Stuttard, Marcus Pinto, Seo Core. The Web Application Hacker's Handbook 2nd: Finding and Exploiting Security Flaws [M]. Wiley, 2011.

[4] Rocha B P S, Conti M, Etalle S, et al. Hybrid Static-Runtime Information Flow and Declassification Enforcement [J]. Information Forensics and Security, IEEE Transactions on, 2013, 8(8): 1294-1305.

[5] Kerschbaum.Simple cross-site attack prevention[C]. In:Proc of Security and Privacy in Communications Networks and the Workshops. 2012:464-472.

[6] Shanmugam J. XSS Application Worms: New Internet Infestation an Optimized Protective Measures [J]. Software Engineering,Artificial Intelligence, Networking,and Parallel/Distributed Computing. 2012: 1164 – 1169.

[7] J.UdhayanandR.Anitha.Demystifying and Rate Limiting ICMP hosted DoS/DDoS Flooding Attacks with Attack Productivity Analysis[C]. Proc. IACC 2012, Patiala.March 2012.

[8] NingChen, Xiao-SuChen , Bing Xionget al.An Anomaly Detection and Analysis Method for Network Traffic Basedon Correlation Coefficient Matrix[C]. Proc. SCAL COM-EMBED DEDCOM' 09, Dalian, Sept.2011.

[9] Xu Rui, Ma Wen-li and Zheng Wen-ling.Defending Against UDP Flooding by Negative Selection Algorithm based on Eigenvalue Sets[C]. Proc. IAS'09, Xi'an,Aug. 2013.

[10] AHN L V, BLUN M, HOPPER N J, et al. CAPTCHA: using hard AI problems for security[C]. Lecture Notesin Computer Science,vol 2656.2014, 12-62

[11] Guha S, Rastogi R, Shim K. CURE: an efficient clustering algorithm for largedatabases[C]. In: Haas LM, Tiwary A, eds. Proceedings of the ACM SIGMOD International Conference on Management of Data. Seattle: ACM Press, 2013, 73-84.