UDC 004.492.3 **DOI: 10.18413/2518-1092-2021-6-2-0-3**

Goncharenko Yu.Yu.
Devitsyna S.N.
SHapovalov P.A.

AUTOMATIC DETECTION OF JAVASCRIPT SNIFFERS

Sevastopol State University, 33 Universitetskaya St., Sevastopol, 299053, Russia

e-mail:iuliay1985@mail.ru, sndevitsyna@sevsu.ru, actek2009@gmail.com

Abstract
The method of automatic detection of javascript sniffers in the code of online stores is considered. As well as its software implementation, which allows you to detect embedded javascript code with a high probability, notify the resource administrator and the user who is on the payment page, thereby protecting the online store from reputational threat.
Keywords: javascript sniffer, online store protection, javascript sniffer detection program.
For citation: Goncharenko Yu.Yu., Devitsyna S.N., Shapovalov P.A. Automatic detection of javascript sniffers // Research result. Information technologies – T.6, №2, 2021. – P. 18-24. DOI: 10.18413/2518-1092-2021-6-2-0-3

Гончаренко Ю.Ю.
Девицына С.Н.
Шаповалов П.А.

АВТОМАТИЧЕСКОЕ ОБНАРУЖЕНИЕ JAVASCRIPT-СНИФФЕРОВ

Севастопольский государственный университет, ул. Университетская, д. 33, г. Севастополь, 299053, Россия

e-mail:iuliay1985@mail.ru, sndevitsyna@sevsu.ru, actek2009@gmail.com

Аннотация
Рассмотрен метод автоматического обнаружения javascript-снифферов в коде интернет-магазинов, а также его программная реализация, которая позволяет с высокой вероятностью обнаруживать встроенный javascript-код, уведомлять администратора ресурса и пользователя, находящегося на странице оплаты, тем самым защищая интернет-магазин от репутационных угроз.
Ключевые слова: javascript-сниффер, защита интернет-магазина, программа обнаружения javascript-сниффера.
Для цитирования: Гончаренко Ю.Ю., Девицына С.Н., Шаповалов П.А. Автоматическое обнаружение javascript-снифферов // Научный результат. Информационные технологии. – Т.6, №2, 2021. – С. 18-24. DOI: 10.18413/2518-1092-2021-6-2-0-3

INTRODUCTION

Online stores are a dynamically evolving online commerce tool. According to forecasts of the Russian research agency Data Insight [Data Insight, 2021], the annual volume of revenue from Internet commerce by 2023 will more than double and will amount to 2.4 trillion. rubles.

The convenience of purchasing goods on the Internet has a downside: buyers who use bank cards to pay online face a variety of cyber threats, one of which is JavaScript sniffers [Technology and Media, 2021; Central Bank of the Russian Federation, 2018; Group-IB, 2021].

Sniffer is a type of malicious code injected by cybercriminals into a victim's website script to intercept user-entered data: bank card numbers, names, addresses, logins, passwords, and so on. When a site is infected, all parties are involved in the chain of victims - end users, payment systems, banks and large companies that sell their goods and services via the Internet.

Having analyzed the existing methods for detecting vulnerabilities in web ap-plications, as well as attack methods [Lukatsky A., 2008; Mark Dowd, John McDonald and Justin Schuh, 2007], a new method for detecting JavaScript sniffers is proposed.

MAIN PART

Analysis of the javascript sniffer detection technique

The technique of automatic detection of javascript sniffers consists in comparing the states of the same page at different intervals according to a certain algorithm. At the initial stage, it is assumed that the web application is "clean", that is, it is not infected with javascript sniffer and other malicious code. The detection system saves a perfect "snapshot" of the page that needs to be monitored for in-fection and puts it in a store of the original state.

An impression represents the following states:

- DOM-tree of the document;
- the counted number of internal and external url-links;
- the weight of each link that has the extension js, gif, png, jpg.

At the necessary time intervals, the system accesses the url page, which is reg-istered, makes its "impression" and compares the current "impression" with the original one. Figure 1 shows the main nodes of the warning system and the inter-action between them.

In case of detection of differences in the states of "snaps" - the notification system about the threat of infection of the url-page being monitored is triggered. A notification about this fact is sent to the resource administrator, and also, if the client of interaction with the browser is connected, to the user's browser window, which is located on the monitored page.

Software implementation of a javascript sniffer detection system

The javascript sniffer detection system consists of the following nodes:

- web-based management interface;
- parsing module;
- module for processing and saving results;
- module for comparison and decision making;
- threat notification module;
- repositories of the initial and current state.

To implement the detection system, the author has chosen the php programming language. Intermediate data will be stored in text files. The advantage of this configuration is that it is cross-platform and easy to deploy.

The web interface is implemented on the bootstrap 4 framework. It consists of two sections - snapshot and monitoring. The monitoring section has an advanced and simple interface. Advanced - for system testing and detailed study of detect-ed threats. Simple interface - combined with the notification module i.e. when accessing the simple interface, information about the presence or absence of a threat is returned without the details that are present in the extended interface.

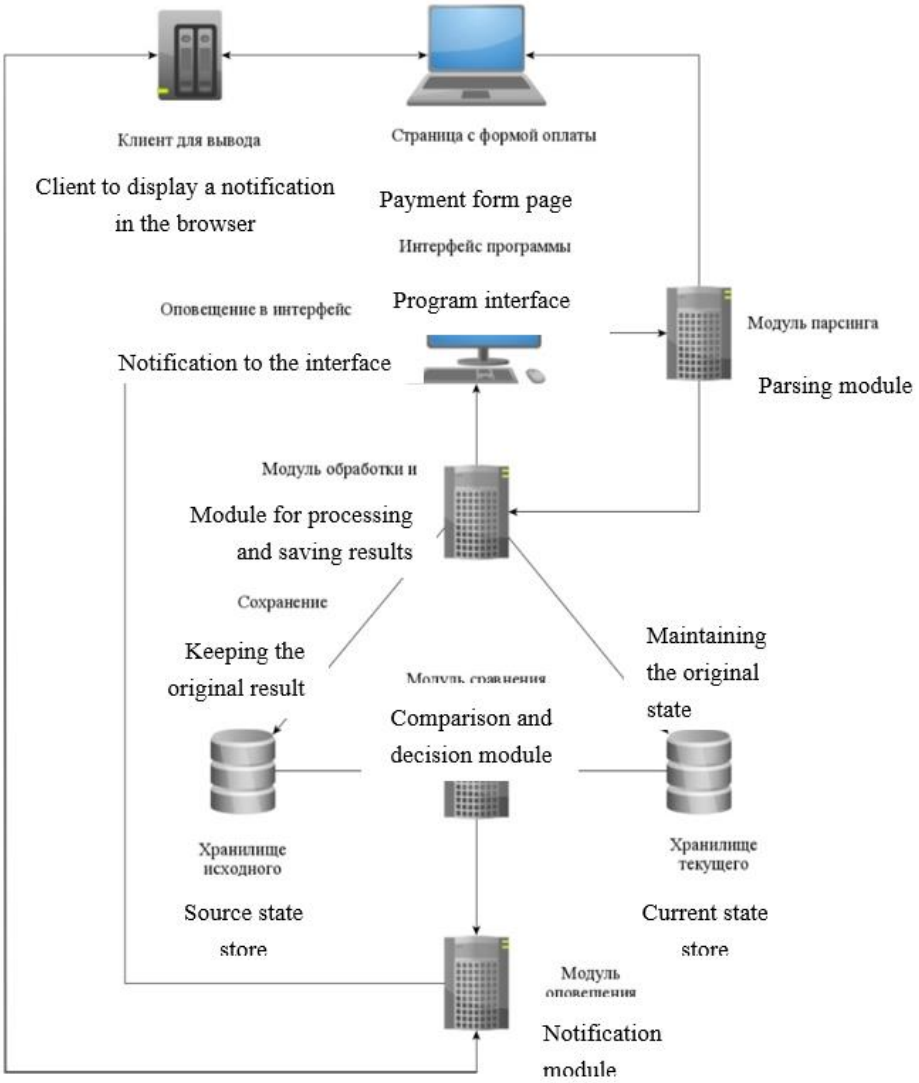

Fig. 1. Nodes of the sniffer notification system

Testing program code

A test page with a form of payment is used to test the scenario of work (see Fig. 2).

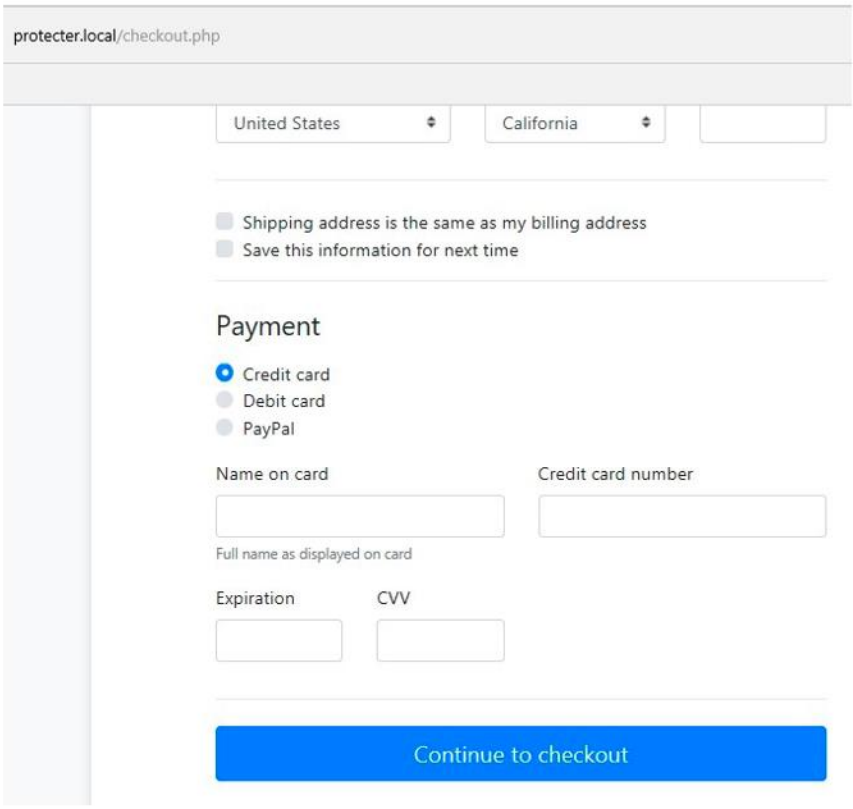

Fig. 2. Test page with payment form

The formed "snapshot" of this page allowed adding it to monitoring (see Fig. 3).

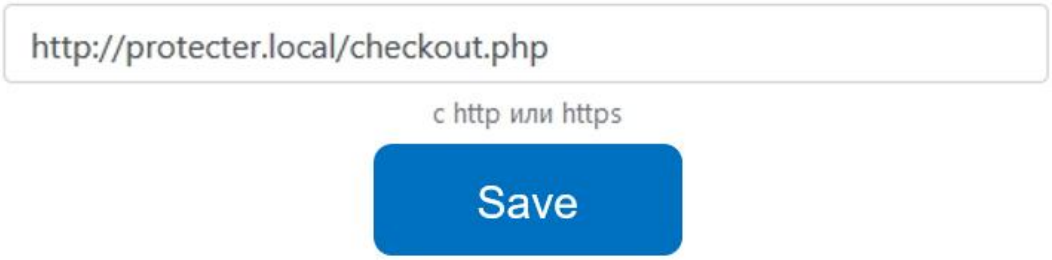

Fig. 3. Procedure for adding a page to monitoring

The monitoring page is a screen divided into two parts (see Fig. 4). On the left, the initial state is displayed, and on the right, the current state. Until threats are detected, the system is in equilibrium.

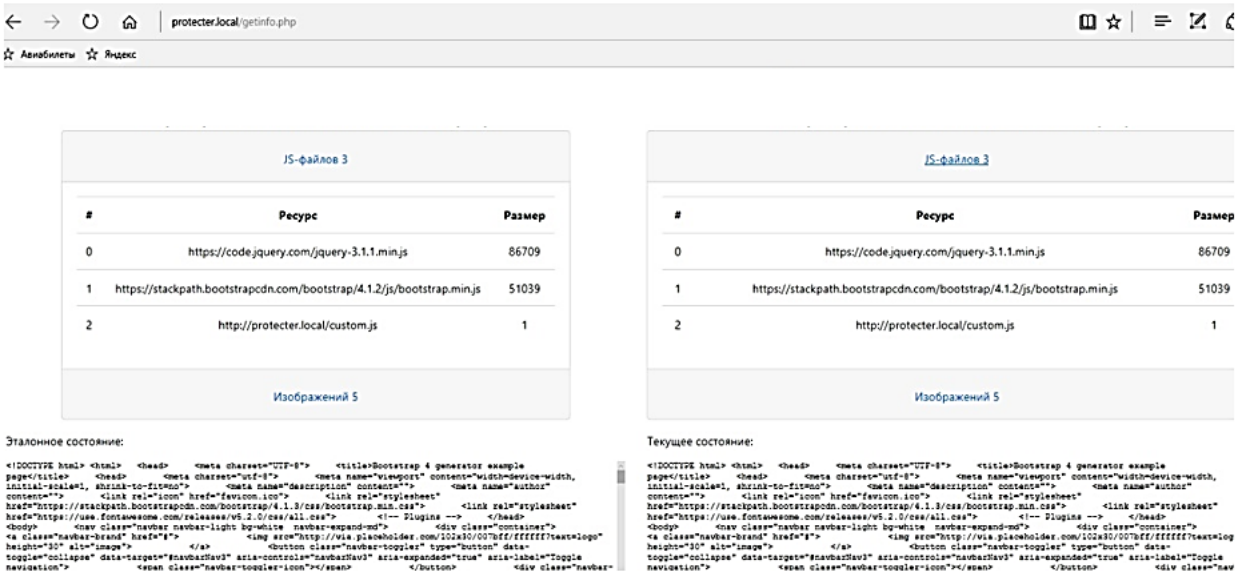

Fig. 4. Advanced interface of the monitoring results page

If you put arbitrary javascript code into a page with a form, a corresponding warning will be displayed in the advanced interface of the detection system (see Fig. 5).

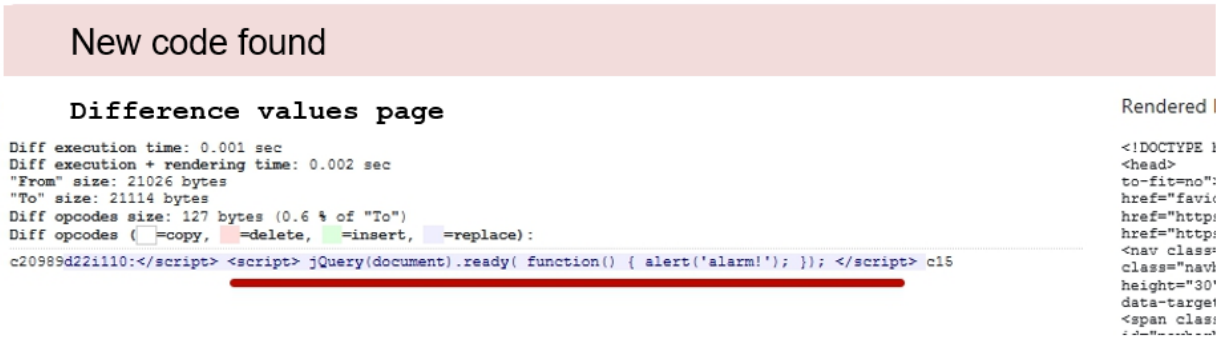

Fig. 5. Notification in the extended interface

And in the browser, the user who is on the page with the payment form will receive a notification about the threat (see Fig. 6).

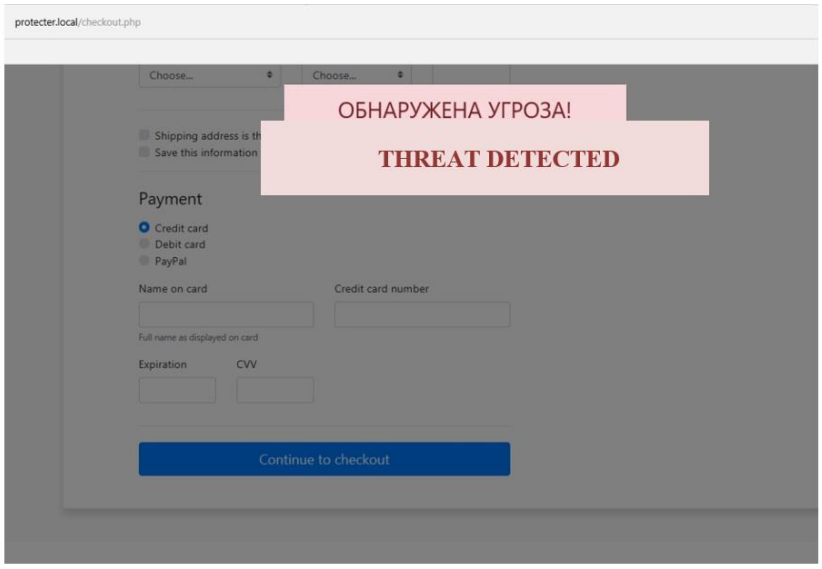

Fig. 6. Notification of a threat in the user's browser

If an attacker tries to place malicious code not directly on the page, but through a file already connected to this page, the system will also notify about this, since the file size will change.

CONCLUSIONS

Thus, using the proposed method for comparing the initial and current state, it is possible to detect the embedded sniffers code with 100% probability, since any change in the code or size of an external file, even by 1 byte, triggers the notification system.

The developed program for detecting changes in the source code can be used:

- to warn about the injection of malicious code into a web application;
- in analytical systems, where it is necessary to track changes in the current state from the initial one;
- an additional module for existing WAFs;
- to compare two texts and find differences.

References

1. Data Insight, 2021. URL: http://www.datainsight.ru/public (date of circulation: 07.05.2021).

2. Technology and Media, 2021. URL: https://www.rbc.ru/technology_and_media, (date of circulation: 07.05.2021).

3. Central Bank of the Russian Federation, 2018. Overview of the main types of computer attacks in the credit and financial sector in 2018. URL: https://cbr.ru/Content/Document/File/72724/DIB_2018_20190704.pdf (date of circulation: 08.05.2021).

4. Group-IB, 2021. Getting to know sniffers: the ReactGet family. URL: https://www.group-ib.ru/blog/reactget, (date of circulation: 09.05.2021).

5. Group-IB. Getting to know sniffers-2: G-Analytics, 2021. URL: https://www.group-ib.ru/blog/g-analytics, (date of circulation: 08.05.2021).

6. Group-IB. Getting to know sniffers-3: Illum, 2021. URL: https://www.group-ib.ru/blog/illum last accessed 2019/06/06 (date of circulation: 08.05.2021).

7. Group-IB. Getting to know sniffers-4: CoffeMokko, 2021. URL: https://www.group-ib.ru/blog/coffemokko (date of circulation: 08.05.2021).

8. Lukatsky A., 2008. Detection of attacks. Textbook. BHV. ISBN: 5-94157-246-8. (In Russian).

9. Mark Dowd, John McDonald, and Justin Schuh, 2007. The art of software security assessment: identifying and preventing software vulnerabilities. Indianapolis, Ind: Addison-Wesley.

Гончаренко Юлия Юрьевна, доктор технических наук, доцент, профессор кафедры «Информационная безопасность» Института радиоэлектроники и информационной безопасности
Девицына Светлана Николаевна, кандидат технических наук, доцент, доцент кафедры «Информационная безопасность» Института радиоэлектроники и информационной безопасности
Шаповалов Павел Анатольевич, главный специалист одела информационной безопасности Аппарата Законодательного Собрания города Севастополя

Goncharenko Yuliya Yur'evna, Doctor of Technical Sciences, Associate Professor, Professor of the Department of Information Security of Institute of Radio-Electronics and Information Security of Sevastopol State University
Devitsyna Svetlana Nikolaevna, Candidate of Technical Sciences, Associate Professor, Associated Professor of the Department of Information Security of Institute of Radio-Electronics and Information Security of Sevastopol State University
Shapovalov Pavel Anatol'evich, Chief Specialist of the Information Security Department of the Legislative Assembly of the City of Sevastopol