GEOGRAPHICAL ASSESMENT OF RESULTS FROM PREVENTING THE PARAMETER TAMPERING IN A WEB APPLICATION

O. Menemencioğlu [a, b, *], İ. M. Orak[a]

[a] KBU, Dept. of Computer Engineering, 78050 Karabük, Turkey - (omenemencioglu, imorak)@karabuk.edu.tr
[b] Directorate of Computer Center, , 78050 Karabük, Turkey

KEY WORDS: Security, Web Application, Web Application Security, Parameter Tampering, Parameter Tampering Attack, SQL Injection, Geography

ABSTRACT:

The improving usage of internet and attained intensity of usage rate attracts the malicious in around the world. Many preventing systems are offered by researchers with different infrastructures. Very effective preventing system was proposed most recently by the researchers. The previously offered mechanism has prevented the multi-type vulnerabilities after preventing system was put into use. The attack attempts have been recorded. The researchers analysed the results geographically, discussed the obtained results and made some inference of the results. Our assessments show that the geographical findings can be used to retrieve some implication and build an infrastructure which prevents the vulnerabilities by location.

1. INTRODUCTION

1.1 Concepts

Development in technology and networking has revealed new areas such as mobile devices, e-commerce, social media, etc. Internet becomes essential part of daily life with these products. The web applications have some vulnerabilities. On the other hand, wide usage of web based applications attracts the malicious attacks because of the vulnerabilities.

Parameter tampering is very important vulnerability for web applications. Web parameter tampering is included by four type of attacks, “Injection”, “Insecure Direct Object References”, “Invalidated Redirects” and “Forwards and Missing Function Level Access Control” (Menemencioğlu & Orak, 2017).

Detecting and preventing tampering attacks, there are three approach, static, dynamic, and hybrid analysis (Menemencioğlu & Orak, 2017). Static approach is based absence of integrity constraint enforcement (Zhang et al., 2011), dynamic approach is processed during execution time by checking web server responses without needing modification of application codes (Menemencioğlu & Orak, 2017). Hybrid approach uses static analysis and provide dynamic approach with runtime detector.

Static approach analyses the code without execution (Natarajan & Subramani, 2012). It focuses on source codes as text and requires rewriting web applications (Lee, Jeong, Yeo, & Moon, 2012).

Dynamic approach analyses the vulnerabilities during execution time (Natarajan & Subramani, 2012). It focuses on instructions in run time (Schwartz, Avgerinos, & Brumley, 2010). It checks web server responses for each input and does not need modification of web application (Lee et al., 2012).

On the other hand, hybrid method simultaneously analyses web pages and generates SQL queries to test (Lee et al., 2012).

The internet usage is extended by the improvement in web security. Furthermore, it effects the trends of commercial and so on (Boyle & Alwitt, 1999). Researches in security field will effect the related preventing security products and they increase the internet usage.

1.2 Background

(Menemencioğlu & Orak, 2017) proposed a detection system for preventing parameter tampering based on the Deterministic Finite State Machine (DFSM) which uses hybrid analysis approach. A technical description is detailed in related research, only a brief description is included here. The detection system is implemented on a faculty information system. The attack attempts are prevented and registered. The registries are accumulated. Figure 1 presents the proposed preventing system.

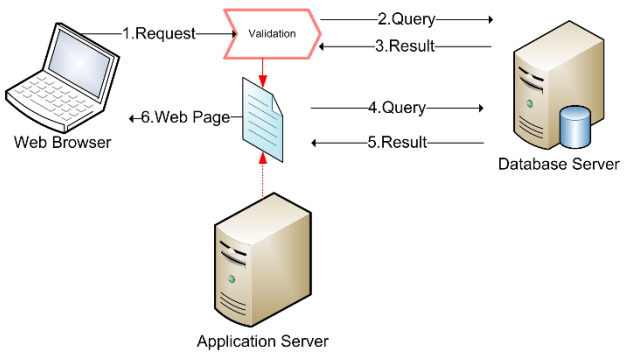

Figure 1. Architecture of implementation.

1.3 Dataset

The accumulated data which consist of prevented attack registries, is analysed in previous research (Menemencioğlu & Orak, 2017). Beyond that study, geographical effects of the proposed system are analysed and discussed in below.

* Corresponding author

Two data sets are held in the study. First goal was detecting attacks. For that reason, first data set covers the period starting from right beginning, includes thirteen-month data. After ensuring detection, the mechanism is enhanced to compare session based data. Second data set has shorter period, but it is much more detailed, covers 6 months of data including session data.

2. EXPERIMENT

2.1 Results

The mechanism that we propose achieves successful results to prevent parameter tampering attacks. Only cost for this operation is the time for checking the parameters on database server. Query time dependents on workload of server. It is not considered here, since it is not the subject of the this paper. If parameter tampering is detected, then there will be time cost for storing attack. The process can be executed in θ (1) time, so time complexity is θ (1).

For detecting tampering attack, the mechanism developed on Karabük University Engineering Faculty is "Academic Curriculum Vitae Based Faculty Information System" (Menemencioğlu, Sonuç, Karaş, & Orak, 2013). This application aims to self-access, manage, manipulate personal data and publish the mentioned entry results for web visitors.

61991 attacks were detected in first dataset. After removing duplicate IP addresses, 3354 unique IP addresses remained. Figure 2 shows the distribution of IP addresses. 98 percent of IP addresses are global IP addresses. Two percent of addresses are NAT IP addresses from Karabük University.

Figure 2. Distribution of IP addresses

Locations of these global IP addresses are identified. Figure 3 shows the country distribution of IP addresses. 42 percent of attacks are detected from Turkey. 23 percent of attacks are detected from United States. 13 percent of attacks are detected from Russia. Fourth and fifth countries are Ukraine and China with about 5 and 3 percent accordingly.

These results also show miss tries. Because some IP addresses are tried more than once. Rest of the IP addresses are not standalone but they are in series. Some examples are listed below.

131.253.24.x (21 IP addresses of 3354 IPs has 0.006 weight, 1%)
180.76.5.x (38 IP addresses of 3354 IPs has 0.011 weight, 1%)
157.x.y.z (x:55-56;y:32-36,229) (166 IP addresses of 3354 IPs has 0.049 weight, 5%)

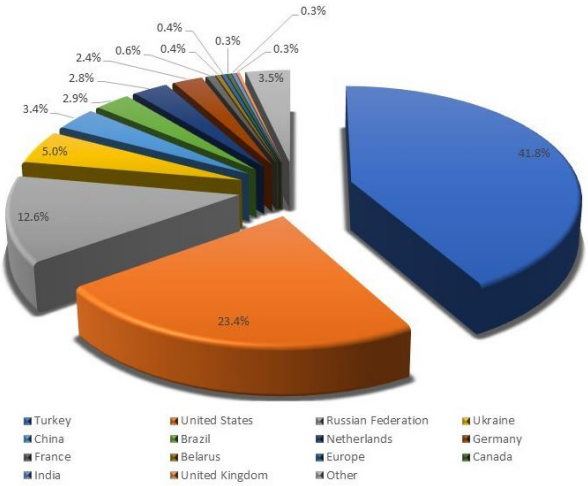

Figure 3. Country distribution of IP addresses

Examples show that these IP addresses are most probably produced by a machine. Because IP address x.y.z.38 is detected after ten days later then the IP address x.y.z.37 is detected. An automatic IP address changer mechanism is detected. Either hardware or software would be used to change current IP address to attack.

The series can be accepted as one IP address. However, an elimination process is very difficult to ignore duplicated IP addresses for same location. Error probability is very high to process. Hence calculations are computed without elimination.

When IP addresses analyzed; 58 percent of them are A class IP address. 31 percent of them are B class IP address. And 11 percent of them are C class IP address. Figure 4 shows the class distribution.

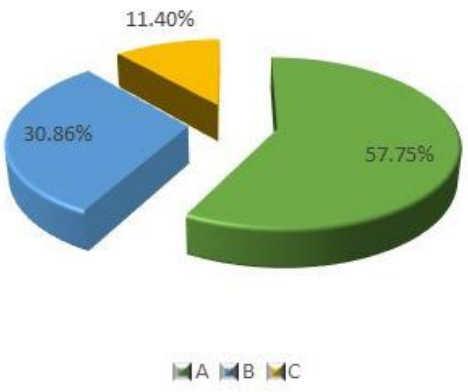

Figure 4. Class distribution of IP addresses

A class IP addresses support large scale of IP addresses. It can be deduced from this that IP changer mechanisms or crawler like mechanisms are possibly used. The distribution of classification support the idea of crawler or IP changer implication.

3. DISCUSSION AND CONCLUSION

In the previous study, the detection system was implemented on a faculty information system. The results are found very successfully when they are compared with other methods in terms of real time detection and evaluation, method, class and code adjustment instead of accuracy. The accuracy comparison

needs same attack dataset to evaluate. In dynamic analysis, providing this type dataset is very difficult. So, accuracy comparison is neglected. The proposed mechanism has simple and effective architecture and implementation which is hybrid analysis based, and very low algorithm overhead. It is cost effective and fastest method in compared. The proposed mechanism attempts to prevent four most common risk types "Injection", "Insecure Direct Object References", "Invalidated Redirects" and "Forwards and Missing Function Level Access Control" which involve parameter tampering (Menemencioğlu & Orak, 2017).

In this research, the accumulated data is examined in detail from geographical point of view.

Implementation web application is in Turkey. Except from Turkey, the most vulnerability attacks are from USA and Russia, Ukraine and China. This can be involved economic and technological development level except Ukraine. In other words, the attackers are mostly located in most developed countries.

Ukraine can be assessed as a splash of the Russian effect when the political impression is considered. Total Europe vulnerability value is about 7 percent.

The distribution of IP series and the rate of A class IP distribution imply the existence of vulnerability detection mechanisms or attack mechanisms like a crawler. Future work will give some thought to these mechanisms.

These appreciations lead the IP based or location based restriction in web based applications, firewall software and hardware products. Future works can focus on IP matching algorithms in the light of the retrieved information in this research.

REFERENCES

Boyle, B. A., & Alwitt, L. F. (1999). Internet Use within the U.S. Plastics Industry. *Industrial Marketing Management*, *28*(4), 327–341. http://doi.org/10.1016/S0019-8501(98)00012-1

Lee, I., Jeong, S., Yeo, S., & Moon, J. (2012). A novel method for SQL injection attack detection based on removing SQL query attribute values. *Mathematical and Computer Modelling*, *55*(1–2), 58–68. http://doi.org/10.1016/j.mcm.2011.01.050

Menemencioğlu, O., & Orak, İ. M. (2017). A Simple Solution to Prevent Parameter Tampering in Web Applications. In *Threat Mitigation and Detection of Cyber Warfare and Terrorism Activities* (pp. 1–20). IGI Global. http://doi.org/10.4018/978-1-5225-1938-6.ch001

Menemencioğlu, O., Sonuç, E., Karaş, İ. R., & Orak, İ. M. (2013). Academic Curriculum Vitae based Faculty Information System. In *XV. Akademik Bilişim Conference Proceedings* (pp. 1123–1127). Antalya, Turkey. Retrieved from http://ab.org.tr/ab13/kitap/menemencioglu_sonuc_AB13.pdf

Natarajan, K., & Subramani, S. (2012). Generation of Sql-injection Free Secure Algorithm to Detect and Prevent Sql-Injection Attacks. *Procedia Technology*, *4*, 790–796. http://doi.org/10.1016/j.protcy.2012.05.129

Schwartz, E. J., Avgerinos, T., & Brumley, D. (2010). All You Ever Wanted to Know About Dynamic Taint Analysis Forward Symbolic Execution (but might have been afraid to ask) A Few Things You Need to Know About Dynamic Taint Analysis Forward Symbolic Execution (but might have been afraid to ask) The Root. *Policy*, 1–5.

Zhang, H., Kuan Tan, H. B., Zhang, L., Lin, X., Wang, X., Zhang, C., & Mei, H. (2011). Checking enforcement of integrity constraints in database applications based on code patterns. *Journal of Systems and Software*, *84*(12), 2253–2264. http://doi.org/10.1016/j.jss.2011.06.044