IJCSE International Journal of Computer Sciences and Engineering Open Access

Research Paper Vol.-6, Issue-6, June 2018 E-ISSN: 2347-2693

KMPS: A Hybrid Algorithm to Detect Web Application Vulnerabilities

Komal[1*], S. Deswal[2]

[1] CSED, Deen Bandhu Chhotu Ram University of Science & Technology, Sonipat, India
[2] CSED, Deen Bandhu Chhotu Ram University of Science & Technology, Sonipat, India

**Corresponding Author: sharma.komal8847@gmail.com*

Available online at: www.ijcseonline.org

Accepted: 15/Jun/2018, Published: 30/Jun/2018

Abstract— With the rapid growth of internet all type of services are available online to decrease the user efforts and to make every task easy. A variety of web applications are available for these activities. Web applications contain confidential data of organizations and databases or other information sources. It can be attacked by attackers or hackers, if there are any vulnerabilities present in the web application. So, there is a need to implement security approaches and algorithms to detect the web vulnerabilities. This paper presents a Hybrid algorithm to detect web application vulnerabilities. The proposed hybrid algorithm, KMPS is a combination of Sunday search algorithm and KMP string matching algorithm. KMPS consists of shifting steps and matching steps to detect the attacks and is compared with the existing BM pattern matching algorithm. The results show that the proposed algorithm performs better than BM pattern matching algorithm in context of searching time, accuracy and throughput.

Keywords— Web application security, SQLi, Cross-site scripting, Cross-site request, forgery, buffer overflow, KMPS hybrid algorithm.

I. Introduction

With the increasing internet technologies web applications have become an important and useful aspect of our daily life activities such as online banking, business, education and online shopping etc. The progress of web services and web applications also pose security issues and vulnerabilities. To identify the growing vulnerabilities there is a need to recognize the new vulnerabilities, drawbacks of previous security systems and to present new intelligent security approaches. OWASP (Open Web Application Security Project) listed some of the vulnerabilities as more critical like SQL injection, XSS, CSRF and buffer overflow etc. [1].

Security vulnerabilities are generally the result of programming code errors. These vulnerabilities can be defined as the flaws, an attacker can exploit to take unauthorized access of a system for his/her personal use [2]. SQLi is an injection type vulnerability in which SQL commands are inserted into input data to gain the unauthorized access to web applications DB [1]. In XSS type of attacks an attacker can insert hidden code scripts into URLs and web pages, can make changes in HTML code, cookies and URLs. This exploits the confidentiality of the web applications [1]. When a doubtful web application initiates irregular activity by a user's web browser on a trusted web application, it is called as cross-site scripting attack [1]. Buffer overflow is an action in which memory assigned to a specific web application becomes massive, attacker's activity affects the memory of web application so that other users cannot use web application's services properly [2]. XML injection, XPath injection, Path traversal and Denial of service are some other popular web application vulnerabilities. False positive and false negative are general problems produced by many detection techniques while discovering the vulnerabilities in a web application. False positive is a situation in which the scanner by mistake finds vulnerabilities in web applications when in reality there is none and when the scanning tool does not find any vulnerability and declares the web application as secure even when there are vulnerabilities present in the web application called as false negative [2].

In this paper, a Hybrid KMPS algorithm is proposed to detect web application vulnerabilities. Matching step from KMP string matching algorithm and shifting steps from Sunday search algorithm are taken to form this hybridized form algorithm. This algorithm compliments the BM string searching algorithm and effectively improves searching time, accuracy and throughput of detection process.

The paper is structured as follows, Section I describes introduction, Section II describes the related work for web application vulnerability detection, section III explains the proposed approach, section IV represents the results & evaluations and finally section V ends with conclusion and future work.

II. Related Work

Fuqiang Yu [3] has proposed an algorithm based Boyer-Moore algorithm for suspicious URL detection. In his work URLs were validated by anti-virus scanners and the accuracy was more than 90 percent. Mahmoud et al. [4] presented an analysis of XSS vulnerability detection. The protection techniques were presented which described many social media web applications XSS vulnerabilities and techniques used to scan and protect the web application from vulnerabilities. Marashdih & Zaaba [5] have proposed XSS vulnerability detection and prevention approach for PHP web applications. They have used idea of GA generators. Gupta [7] has used word segmentation and BM algorithm for malicious domain detection in web applications and described URL components. His approach was 85 percent more accurate than other anti-virus scanning tools. A hybrid algorithm based on Cuckoo search and Artificial Bee Colony algorithm is proposed by Prashanth et al. [8] for vulnerability scanning and cloud security. Yu, Tao & Lin [9] and Muiruri et al. [10], have proposed hybrid algorithms to discover the invasions and vulnerabilities in the web applications. Patel & Shekokar [11] have proposed a string matching approach by modifying Aho-Corasick algorithm to protect web applications from SQL injection vulnerabilities. Qiao & Zhang [13] have proposed an improved BM string matching algorithm for invasion discovery systems, 2-dimensional string matching algorithms was compared by Chang & Wang [21] which considered KMP and Rabin-Karp algorithms and proposed a new KMP+Rabin-Karp algorithm. Another hybrid string matching algorithm of Boyer-Moore and KMP algorithm have proposed by Xian-feng et al. [22].

III. Methodology

In this proposed algorithm, the existing KMP string matching and Sunday search algorithms are hybridized to detect the SQL injection, Cross-site scripting, Cross-site request forgery and Buffer overflow web application vulnerabilities. The algorithm uses the shifting logic of Sunday search algorithm and matching logic of KMP algorithm to detect the malicious pattern within an URL. BM (Boyer Moore) is also a string matching algorithm which starts pattern matching from right to left by applying two heuristics, first is bad character shift and second is good suffix shift to decide the accurate shift distance [2,3]. Bad character shift begins the matching between pattern "P" and text string "X", when a mismatch occurs the algorithm will jump to an "M" (pattern length). After that good suffix shift begins matching from right to left as well and when there is a match the algorithm will jump to the forward character in text sting "X" with next character in pattern "P" [2]. KMPS hybrid algorithm starts searching for Pa=Pa [1….x] in T=T [1…k] by moving Pa from left to right along T. Initially for every step, location m = 1 of Pa is aligned with a location n ∈ 1..k-x+1 in T so that location x of Pa is aligned with location n'= n +x−m in T. While KMP match, pa is matched from left to right with T, until when a match or mismatch is detected at any location of m ∈ 1…x. KMPS algorithm is based upon 2 factors:

1. When a mismatch of Pa pattern with T text string is discovered, Sunday shift is applied to define the next location n' (index value in text string T) T[n']=Pa[x]. In case n' is discovered KMP matching steps are applied.

2. When match of Pa pattern in text string T is detected, KMP matching is continued on Pa[1…x].

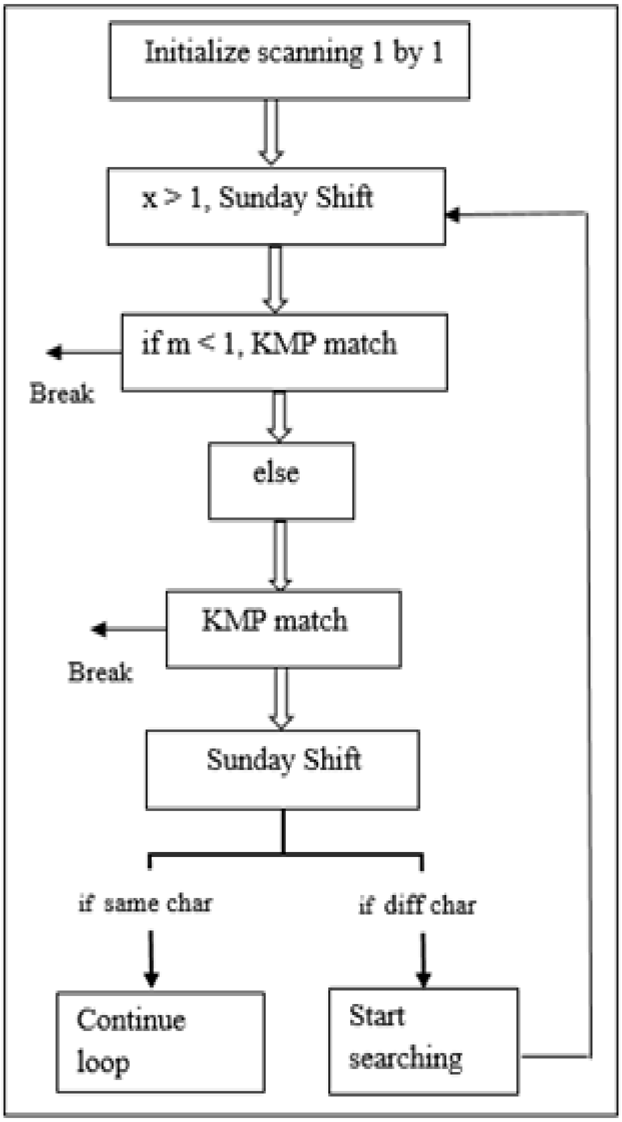

Figure 1: Steps of KMPS hybrid algorithm.

Hybrid algorithm is as following:

```
Pa= Pattern string to be searched.
T= Text string in which pattern string is to be searched.
m= Index of Pa.
n= Index of T.
x'= Any keyword values of Pa.
n'= gives exact position of Pa keywords in T.

Search all occurrences of $p_a = p_a$ [1….x] in T=T [1…k]
if m < 1 then return
m ← 1; n ← 1; n' ← x; x' ← x-1
while n' ≤ k do
if m ≤ 1 then
-When no partial match of $p_a$, apply Sunday shifts,
-Returning next location n' such that T [n']= $p_a$[x]
SUNDAY SHIFT (n')
-Rearrange invariants for KMP match of $p_a$ [1…x-1]
j ← 1; n ← n'-x'
KMP-MATCH(x'; m, n)
else
-Continue KMP match for $p_a$ [1…x]
KMP-MATCH(x; m, n)
-Restore invariants for succeeding attempts
(SUNDAY/KMP)
m ← $\beta$'[m]; n' ← n + x-m
KMP-MATCH(x; m, n)
while m ≤ x and T[n]= $p_a$[m] do
n ← n+1; m ← m+1
if m > x then output n - x
SUNDAY-SHIFT (n')
while T[n'] ≠ $p_a$[x] do
n' ← n'+$\Delta$[T[n'+1]]
if n' > k then return
```

Figure 2: KMPS hybrid algorithm pseudocode.

IV. RESULTS AND DISCUSSION

The proposed KMPS algorithm is compared with BM algorithm for vulnerability detection for searching time, accuracy and throughput. KMPS hybrid algorithm shows better results than BM string matching algorithm in terms of searching time, throughput and accuracy. KMPS is examined on more than 120 URLs, 30 above URLs for each type of vulnerability. It takes less searching time than BM algorithm and its throughput values are also better.

a.) Searching Time: The searching time is the time required to detect the vulnerability, it is measured in nano-seconds (ns). The results are collected for more than one hundred twenty URLs for SQLi, XSS, CSRF and Buffer overflow. KMPS hybrid algorithm takes less time to detect the vulnerability when compared to BM algorithm.

Table 1: Searching Time comparison for KMPS and BM algorithms.

Type	BM	KMPS
SQLi	264814477.6	127342594
XSS	160115073.5	153023721
CSRF	118802929.7	93943918.8
BufferOverflow	79645243.3	74688529.5

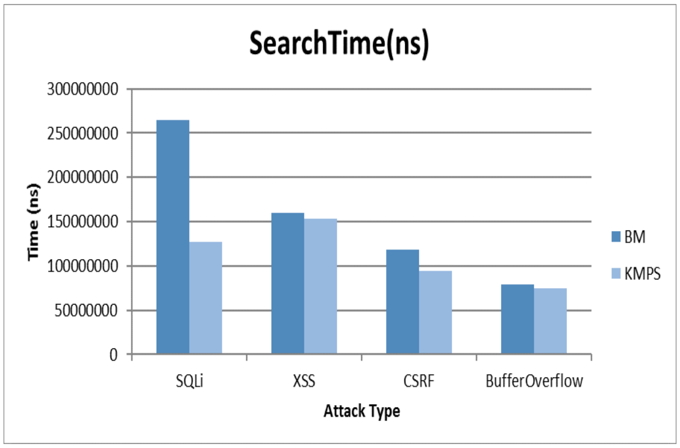

Figure 3: Vulnerability Searching time taken by BM and KMPS hybrid algorithms.

b.) Average Throughput: The throughput is defined as the number of vulnerabilities detected in a unit of time. For all SQLi, XSS, CSRF and Buffer overflow vulnerabilities, the average throughput is calculated as:

Avg. Throughput = total number of detected vulnerabilities/ time taken for detection (ns).

Table 2: Throughput comparison for KMPS and BM algorithms.

Type	BM	KMPS
SQLi	11.82879	20.84
XSS	14.89	16.425
CSRF	27.2	30.5
BufferOverflow	29.25	30.5

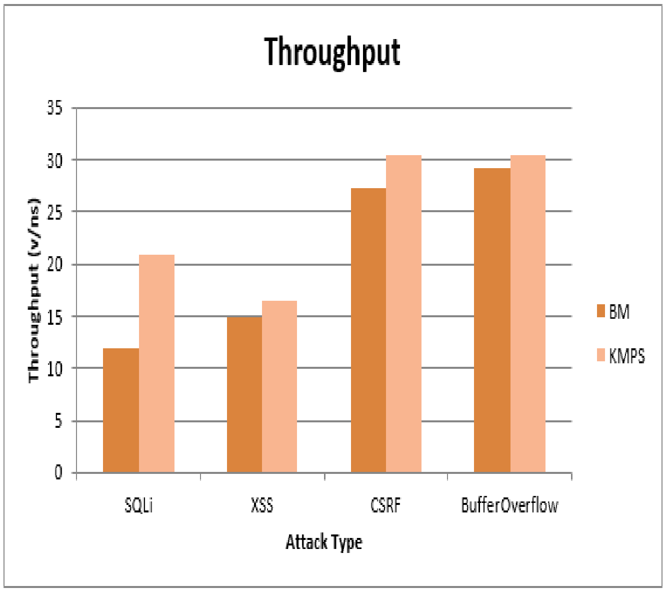

Figure 4: Average Throughput analysis of BM and KMPS hybrid algorithms.

c.) **Accuracy:** The accuracy is calculated on the basis of false positives and false negatives rate occurrence.

Accuracy= (No. of false positive or false negative occurrences/No. of correct occurrences) ×100.

Table 3: Accuracy comparison for KMPS and BM algorithms.

Type	BM	KMPS
SQLi	70.9	72.8
XSS	83.2	86.1
CSRF	88.5	89.7
BufferOverflow	85.4	87.6

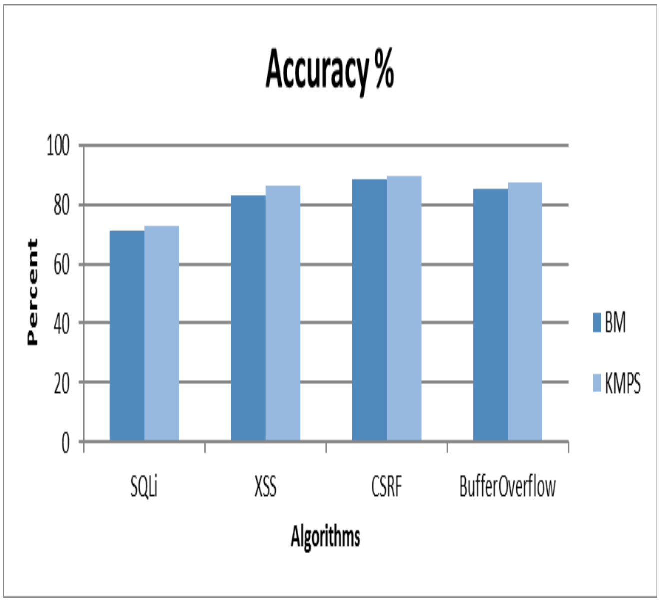

Figure 5: Accuracy percentage of BM and KMPS hybrid algorithms.

V. Conclusion and Future Scope

In this paper a Hybrid KMPS string matching algorithm is proposed, which uses both KMP string matching algorithm and Sunday search algorithm steps to detect the web application vulnerabilities. These vulnerabilities cannot be completely removed but at every stage of development various algorithms and testing methods can be applied to detect vulnerabilities. After examining the results and a series of test data, it is concluded that KMPS algorithm has better accuracy rate, searching time and throughput than the existing BM algorithm. Further, this algorithm can be used to detect other vulnerabilities also, in addition to SQLi, XSS, CSRF and Buffer overflow.

References

[1] Nagpal, B., Chauhan, N., & Singh, N. (2017). "SECSIX: security engine for CSRF, SQL injection and XSS attacks." International Journal of System Assurance Engineering and Management, 8(2), 631-644.

[2] Saleh, A. Z. M., Rozali, N. A., Buja, A. G., Jalil, K. A., Ali, F. H. M., & Rahman, T. F. A. (2015). "A method for web application vulnerabilities detection by using boyer-moore string matching algorithm." Procedia Computer Science, 72, 112-121.

[3] Yu, F. (2015). "Malicious url detection algorithm based on bm pattern matching." International Journal of Security and Its Applications, 9(9), 33-44.

[4] Mahmoud, S. K., Alfonse, M., Roushdy, M. I., & Salem, A. B. M. (2017, December). "A comparative analysis of Cross Site Scripting (XSS) detecting and defensive techniques." In Intelligent Computing and Information Systems (ICICIS), 2017 Eighth International Conference on (pp. 36-42). IEEE.

[5] Marashdih, A. W., & Zaaba, Z. F. (2017, October). "Detection and Removing Cross Site Scripting Vulnerability in PHP Web Application." In Promising Electronic Technologies (ICPET), 2017 International Conference on (pp. 26-31). IEEE.

[6] Thomé, J., Shar, L. K., Bianculli, D., & Briand, L. (2017, May). "Search-driven string constraint solving for vulnerability detection." In Software Engineering (ICSE), 2017 IEEE/ACM 39th International Conference on (pp. 198-208). IEEE.

[7] Gupta, S. (2016, December). "Efficient malicious domain detection using word segmentation and BM pattern matching." In Recent Advances and Innovations in Engineering (ICRAIE), 2016 International Conference on (pp. 1-6). IEEE.

[8] Prashanth, S. K., Rao, N. S., & Kumar, C. S. (2016, March). "Hybrid Cuckoo search—ABC algorithm based vulnerabilities mapping and security in clouds." In Electrical, Electronics, and Optimization Techniques (ICEEOT), International Conference on (pp. 2569-2572). IEEE.

[9] Yu, J., Tao, D., & Lin, Z. (2016, August). "A hybrid web log based intrusion detection model." In Cloud Computing and Intelligence Systems (CCIS), 2016 4th International Conference on (pp. 356-360). IEEE.

[10] MUIRURI, C. K., Ruhiu, S., & Moturi, C. A. (2015). "A HYBRID ALGORITHM FOR DETECTING WEB-BASED APPLICATIONS VULNERABILITIES."

[11] Patel, N., & Shekokar, N. (2015). "Implementation of pattern matching algorithm to defend SQLIA." Procedia Computer Science, 45, 453-459.

[12] Hazel, J. J., Valarmathie, P., & Saravanan, R. (2015, February). "Guarding web application with multi-Angled attack detection." In Soft-Computing and Networks Security (ICSNS), 2015 International Conference on (pp. 1-4). IEEE.

[13] Qiao, J., & Zhang, H. (2015, September). "Improvement of BM algorithm in intrusion detection system." In Software Engineering and Service Science (ICSESS), 2015 6th IEEE International Conference on (pp. 652-655). IEEE.

[14] Srivastava, M. (2014, March). "Algorithm to prevent back end database against SQL injection attacks." In Computing for Sustainable Global Development (INDIACom), 2014 International Conference on (pp. 754-757). IEEE.

[15] Trinh, M. T., Chu, D. H., & Jaffar, J. (2014, November). "S3: A symbolic string solver for vulnerability detection in web applications." In Proceedings of the 2014 ACM SIGSAC Conference on Computer and Communications Security (pp. 1232-1243). ACM.

[16] Kadhim, H. A., & AbdulRashidx, N. (2014, June). "Maximum-shift string matching algorithms." In Computer and Information Sciences (ICCOINS), 2014 International Conference on (pp. 1-6). IEEE.

[17] Razzaq, A., Anwar, Z., Ahmad, H. F., Latif, K., & Munir, F. (2014). "Ontology for attack detection: An intelligent approach to web application security." Computers & security, 45, 124-146.

[18] Shar, L. K., Tan, H. B. K., & Briand, L. C. (2013, May). "Mining SQL injection and cross site scripting vulnerabilities using hybrid program analysis." In Proceedings of the 2013 International Conference on Software Engineering (pp. 642-651). IEEE Press.

[19] Ding, S., Tan, H. B. K., Shar, L. K., & Padmanabhuni, B. M. (2013, December). "Towards a Hybrid Framework for Detecting Input Manipulation Vulnerabilities." In Software Engineering Conference (APSEC), 2013 20th Asia-Pacific (Vol. 1, pp. 363-370). IEEE.

[20] Lu, C. W., Lu, C. L., & Lee, R. C. (2013). "A new filtration method and a hybrid strategy for approximate string matching." Theoretical Computer Science, 481, 9-17.

[21] Chang, C., & Wang, H. (2012, March). "Comparison of two-dimensional string matching algorithms." In Computer Science and Electronics Engineering (ICCSEE), 2012 International Conference on (Vol. 3, pp. 608-611). IEEE.

[22] Xian-feng, H., Yu-bao, Y., & Lu, X. (2010, August). "Hybrid pattern-matching algorithm based on BM-KMP algorithm." In Advanced Computer Theory and Engineering (ICACTE), 2010 3rd International Conference on (Vol. 5, pp. V5-310). IEEE.

[23] Shreekishan Jewliya, "Analysis of Web Application Security", International Journal of Computer Sciences and Engineering, Vol.5, Issue.9, pp.215-220, 2017.

[24] Sandeep D Sukhdeve and Hemlata Channe, "A Survey on Content Injection Attacks", International Journal of Computer Sciences and Engineering, Vol.3, Issue.11, pp.70-74, 2015.

Authors Profile

Suman Deswal is pursuing her Ph.D from DCR University of Science & Technology, Murthal, India. She has completed her M.Tech (Computer Science & Engineering) from Kurukshetra University, Kurukshetra, India and B.tech (Computer Science & Engineering) from CR State College of Engineering, Murthal, India in 2009 and 1998 respectively. She has published more than 30 papers in various international journals.

Komal is persuing her M.tech from DCR University of Science and Technology, Murthal, India. She has completed her B.tech (IT) from Bhagat Phool Singh Mahila Vishvavidyalaya, Khanpur Kalan (Sonipat), India in 2016.