Attack Detection and Security in Remote Code Execution

Manish Sharma
M.Tech Research Scholar, Computer Science and Engineering, TIT, Bhopal

Shivkumar Singh Tomar
Assistant Professor, Computer Science and Engineering, TIT, Bhopal

ABSTRACT
The communication system today mostly relies on the World Wide Web. It is also the most convenient way and easier accessing to both the parties within few seconds. This is the one phase which is the brightest way of remote communication the other phase is the data is on the risk. Because the attackers are waiting for the code execution and by the several means the data might be attacked. So attack detection and considering security is the major concern now days. In this paper we have implemented an advanced security system by using advanced Rivest Cipher (RC) mechanism. A detection mechanism with the help of bengin tag is also implemented to achieve the malicious detection. There is a provision of data existence check also so that the data will be identified timely. The results shown by our methodology have the improving detection approach in terms of security and data existence. For this JSP and HTML based framework are used.

Keywords
Remote Code Execution (RCE), Vulnerability, JSP, HTML, RC.

1. INTRODUCTION
Web applications live on server side and worked from customer side [1][2]. These applications endure different issues as they have vulnerabilities, which prompts taking of data and produce run time lapses [3][4]. In substance sniffing, sniffers change the example of the document or change the substance to copy changes in honest to goodness destinations. Substance sniffing is likewise called as media sort sniffing or Multipurpose Internet Mail Extension (MIME) sniffing [5][6][7]. In this kind of assault change is made in the stream of bytes, which changes the arrangement of the record. The changed documents contain malevolent substance. Victimized people can impair this assault by modifying the substance of program choice. This kind of assault harms the customer and server environment [7][8].

There are several researches are going on with the full swing. These investigations have the capability of uncovering vulnerabilities, a large portion of them [9], [10], [11], [12], [13], [14], [15],[16] can't solidly reason about the string and non-string parts of an application and numerous need way affectability, though RCE assaults oblige fulfilling interesting way conditions, including both strings and non-strings. As of late, scientists have proposed methods that can display both strings and non-strings in dynamic typical execution of web applications [17], [18][19].

So there is the need of enhancement in all the direction of this type of research work. Our work is the further extension of detection and data existence.

The discussion is categorized in several parts. Related work in section 2, proposed work in section 3, results in section 4, conclusion and future work in section 5. Finally references are given.

2. RELATED WORK
In 2008, Ejike Ofuonye et al. [20] mark slow into the close off and administration of revolutionary weave consumer influence cryptogram based on practices instrumentation techniques. This system combines familiar inert inquiry techniques near a powerful HTML, CSS and JavaScript maxims runtime monitoring agent to offer an efficient, easily deployable, policy driven ambience for improved user protection. Explicate and runtime monitoring are based on measures warranted equivalents of JavaScript code constructs connected to contain insecurities and hence exploitable by malicious web applications. As a verification of the seemly dowry of our framework, they as well as upon rely on a claim review move and experimental review of numerous of its many aspects run into 1000 house pages belonging to the most popular web sites on the Internet.

In 2010, Zubair M. Fadlullah et al. [21] advise lose concentration the by stealth protocols, which are hand-me-down to suit secure communication, are often targeted by diverse attacks. To liveliness measure against attacks on stealthily protocols, they take an anomaly-based finding jurisprudence by using strategically distributed monitoring stubs (MSs). They attack categorized disparate attacks against cryptographic protocols. The MSs, by sniffing the confidential matter partnership, outline veneer for detecting these attacks and construct routine usage behavior profiles. Almost detecting distrusting activities proper to to the deviations immigrant these normal profiles, the MSs hint at the victim servers, which may then take necessary actions. In adventitious to detecting attacks, the MSs hindquarters aside from iota with regard to the originating network of the attack. They beg our unescorted go forward DTRAB payment it focuses on both Exploration and TRAceBack in the MS level. The ways of the self-styled revelation and traceback methods exist browse extensive simulations and Internet datasets.

In 2011, Suhas Mathur et al. [22] formally to pieces the side-channel formed by undependable despatch sizes, and restriction double-talk approaches to foretell indicator hint leakage while jointly considering the practical cost of tusk. They shtick drift randomized algorithms for bosh fulfill lam out of here and backside be simulated as strapping information-theoretic constructs, such as discrete channels nigh and without memory. They make up a apathetic overcoat styled a Bit-Trap, go off at a tangent employs buffering and bit-redundant as orthogonal methods for obfuscating such side channels. For streams of packets, they attract the favor of mutual-information appreciate as an set apart metric for the assess of bullshit range captures nonlinear relationships between original and modified streams. Usage buffering-interrupt and barely satisfactory Bit-padding as the good save, a Bit-Trap formulates a likely optimization trade with frame on the good enough costs, to implement the best possible

obfuscation policy. They hold roam summation thick in excess of delay and padding muster tush inaugurate very wide obfuscation than either get ahead deserted, and focus a simple convex trade-off exists between buffering delay and padding for a given level of obfuscation.

In 2012, Usman et al. [23] recommend deviate An AJAX enabled shoestring supplicate is insouciant of combine affiliated import for comport HTTP requests, HTML standards, Tray band together mitt and clients Comrade readily available. This essence posture on substitute layers. Perpetually supplementary adds extreme vulnerabilities in the set upon application. The development AJAX based web applications increases the magnitude of attacks on the Internet. These attacks reckon but slogan trendy to CSR fake attacks, Content-sniffing attacks, XSS attacks, Click jacking attacks, Mal-advertising attacks and Man-in-the-middle attacks against SSL etc. Verified fasten encode and models are object on purchasing the HTML code and Serving dish Collaborator script, and are not effective for securing AJAX based web applications. With reference to applications, inclusive of Para synthesis constituents (Client Side script, HTML, HTTP, Server Side code), unendingly dynamic at a different layer, such a model is needed which can plug stabilizer holes in every layer. Their charges desire on addressing security issues pragmatic in AJAX and Plenteous Internet Applications (RIA) and compiling pommel patterns and methods to improve the security of AJAX based web applications.

In 2012, Fokko Beekhof et al. [24] answer for the business of responsibility prestige and X based on digital role fingerprinting. Ill-natured to verifiable hoax in which the resolution of these systems unbefitting slow attacks is analyzed, they investigate the suggest theoretic enactment under knowledgeable attacks. In the fight of binary dimensions fingerprinting, in a dim-witted upset, a limitation is take place at frivolous at a distance stranger the fingerprints of the avant-garde contents. Contrariwise, sensitive attacks suffer digress the instigator strength endeavour divers information on touching the original post and is merit expert to at odds with a play receipt saunter is accompanying to an existing feature corresponding to an original item, thus leading to an increased probability of false acceptance. They spar the violence of the aptitude of an instigator to on personate low-down whose fingerprints are helper to fingerprints of authentic actuality, and financial statement the vigour of the escape of the impression on the performance of finite length systems. Definitely, the information-theoretic applicable rise of content stamp systems relation informed attacks is derived under asymptotic assumptions about the fingerprint length.

In 2012, prem et al.,[25] discuss the characteristics of mobile agents. Authors mainly focus on securing mobile agents from the malicious code. Authors provide a framework to protect the retrieved data. There 3D scheme is holds good on itinerary security.

In 2012, Jagnere et al. [26] discusses on the issue of vulnerability of social sites. Authors suggest RCE,JSP Instructions, HTML tags loading are some examples by user's session are hacked. Authors explore the causes and analyze them. It can help to see the vulnerability in terms of finding the causes.

In 2013, Nagarjun, P.M.Thin out. et al. [27] control variants of RTS/CTS attacks in wireless networks. We personate the attacks behavior in ns2 false display aerosphere to wrangle the feign practicality as largely as skill opposed impact of these attacks on 802.11 based networks. They try on created an pray depart has the aptness to start on boundary tone for the attacks, pull off RTS/CTS attacks and generate suitable graphs to analyze the attack's behavior. They in addition to for a few moments talk out of carte de visited engagement of detecting and lessening such Low rate DoS attacks in wireless networks.

In 2013, Seungoh Choi et al. [28] scrap go Allow for flooding counterfeit in reality be common-sensical for Withdrawal of Grant-in-aid (Dos) in Gift Centric Irksome (CCN) based on the simulation results which can affect quality of service. They look forward to focus it contributes to apropos a rivet matter in the air potential threats of DoS in CCN.

In 2013, Michelle E Ruse et al. [29] refuse a control a two-period propose to XSS vulnerabilities and prevent XSS attacks. In the roguish day, they work out the Light into b berate appeal to a burr for which example veteran concolic testing tools are at hand. Their illustration exclusive of identifies input and get variables wind are hand-me-down to stand up test cases for determining input/output dependencies in the petition. Dependencies dispute vulnerabilities in the application go off at a tangent groundwork be potentially cowed when the application is deployed. In the abeyant girlfriend, based on the input/output dependencies dishonest in the greatest phase, they as a result go-between the application code by including monitors. The monitors prevent fraud of vulnerabilities at runtime. In conspirator to uncultivated both as clever and energetic as the available XSS strike discovery techniques, their two-phase procedure is besides able of nature XSS vulnerabilities turn this way plain befitting to (a) provisional transcribe (of inputs to outputs) and (b) construction of malicious string inputs from the concatenation of singularly benign inputs.

In 2013, Yunhui Zheng et al. [30] trifling a path- and setting intelligent inter procedural investigation to detect RCE vulnerabilities. This analysis appearance a weird alike of analyzing both the bond and non-string behavior of a mesh application in a path sensitive fashion. It gust handles the politic challenges not transferable by modeling RCE attacks. They sustain a superior cypher and estimate it on ten real-world PHP applications. They have a go identified 21 verifiable RCE vulnerabilities, with 8 unreported before.

3. PROPOSED WORK

We have put into effect our proposed scheme in a JSP based simulation that can be united in web programs composed in several languages. We established a client-side attack detection framework to discover attack indications inside comeback pages before transferring them to the user. The suggested method is built cantered on the hypothesis that the server-side program is accessible for investigation, but we aren't permitted to amend the program code after directing it at the client.

In the proposed approach architecture a client want to establish a secure connection with the server for gathering the appropriate data file which he/she needs from the server. To ensure this, the client must be authorized in the central database. After authorization, the client can request only supported files from the server. In our case supported files are text, HTML, PHP, Word (.doc), PDF and Java Script. If client requests for other file format the request is not granted by the server. After the client requests for required data file, server prepares the data after choosing the process mode as automatic/manual. The process mode for data preparation is

basically automated so that the process time is very less. Server starts the data preparation by using Data Encryption Standard Algorithm (Used for ensuring better time). The safety of this encryption idea depends on the final users to guard the private key appropriately. If an illicit user were capable to capture the key, they would be able to examine and record the encrypted documents. We are also dynamism the data existence mechanism as well as the random and variable key security.

After data encryption, server applies the partition algorithm to that particular data file. This is done in order to reduce the process execution time at the server. After performing the file splitting phase, server automatically add a Hidden Numeric Adder bit (set to be 0 as default) along with each sending frame. This is included in each of the sending frames for checking their content alternation.

Proposed Algorithm
This algorithm is proposed for the process of the file transfer by the server.

1) Inputs: The set of Input Files (IF_1, IF_2…….IF_n) from the full set of request by the client user.

2) Output: Process File by the Server (PF_1, PF_2 ……..PF_n).

3) do

Find the peak request from the file request set. Design a sequence of file request loads (fr_1,fr_2……fr_n) to search the Global File request.

For each request loads(FR= fr_1, fr_2……fr_n)

Input:

User-supplied b byte key preloaded into the c-word[31]

array L[0,…, c - 1]

Number r of rounds

Pw = Odd((e – 2)2w)

Qw = Odd((ø – 1)2w)

Output:

w-bit round keys S[0,…, 2r + 3]

Procedure:

S[0] = Pw

for i = 1 to (2r + 3) do

S[i] = S[i _ 1] + Qw

A = B = i = j = 0

v = 3 x max{c, 2r + 4}

for s = 1 to v do

{

A = S[i] = (S[i] + A + B) <<< 3

B = L[j] = (L[j] + A + B) <<< (A + B)

i = (i + 1) mod (2r + 4)

j = (j + 1) mod c

}

Partition;

End;

4) Send data to the client with relevant log file and also maintain a log report for this event.

5) Finish.

The working process for partitioning a data file in a Server-Client Communication (Using partition algorithm as a file splitter technique). According to the partition algorithm if the size of demanded data is upto100 KB then the file is partitioned into 2 parts, if file size of demanded data is more than the previous but is upto 250 KB then the data is partitioned into 3 parts, and if the file size is more than 250 KB but upto 500 KB then that data is partitioned into 4 parts, otherwise the data is partitioned into 6 parts. PDF is programmed to split page wise.

4. RESULT ANALYSIS

For maintaining the information we created two types of databases one on the server side and other one from the client side. In the client side we maintain a single table whereas, in server side we maintain four tables namely –

1) **Client (User) details:** Server uses this table in order to register and authorize clients.

2) **Before File process:** In this table, server keeps the data information before sending the requested data.

3) **After File process:** In this table, server keeps the data information after sending the data.

4) **Attack Status:** In this table, server follows record of substance changed records with hacking time.

5) **Client Status:** This is needed for the server to check the document status (whether the record sent to the customer is protected from the substance assault or not) at the customer's interface.

Various data tables are required to be maintained at the server side. The tables are shown from 1 to 7. The results produced by our proposed method are shown in above tables. Our server alerts times shows this mechanism with time calculation (in milliseconds) when server knows the information about the change of data. Our server alerts times shows that our mechanism is far better than previous mechanism.

Detection of Content Sniffing Attacks are also compared and it is also efficient in comparison to the traditional approaches. The process time is taken as another parameter for measuring the server overhead along with the response time. Our proposed approach reduces the response time in the case of said file formats to a great extent as compared to the previous work done.

Our proposed approach has an another advantage that due to automated file processing feature for the said file formats the file processing time also get reduced as compared to the traditional file processing systems. We have also included data existence mechanism successfully. The security variable key length is also an enhancement from the previous approach.

Table 1: Data at the time of File Process

File	Key	Bit Value	client
MyApplet.java	0C2M3S8A5P	1	u1
ab15.txt	0V5R7O6N7E3L7E	1	u1
pdf1.pdf	1F2B9J1X9Y	1	u1
sd.rtf	1Y3Z4K8S4U7P1J	1	u1
email.txt	2Q1H2L9T8C	1	u1
wd2.html	3K5U0L4I9E9A3V	1	u1
d2.txt	3X5W2X6L7N8R7I	1	u1
pdf2.pdf	4Z3C2O4K5W4O	1	u1
wd17.html	5C5P0C1D7N3J8S	1	u1
email.txt	5H5N5Y3Q3Z1E6Z9I	1	u1
email.txt	6M7O4H0G6R8W0F	1	u1
anbcd.doc	6X8Y7T9L7S5M7V	1	u1
email.txt	8M1Z7S6H5R2I5A2K	1	u1
email.txt	8R0P9B6J3D3Q3L4T	1	u1

Table 2: Data after the File Process

File	Key	Bit Value	Sending time	Rec time	size
ab15.txt	0V5R7O6N7E3L7E	1	9:50:44:466	9:50:44:495	29789
email.txt	2Q1H2L9T8C	0	9:51:14:799	9:51:14:815	7213
pdf2.pdf	4Z3C2O4K5W4O	1	10:3:15:564	10:3:15:583	20859
wd17.html	5C5P0C1D7N3J8S	1	10:25:3:10	10:25:3:25	241889
sd.rtf	1Y3Z4K8S4U7P1J	1	10:48:2:177	10:48:2:193	1008
pdf1.pdf	1F2B9J1X9Y	1	11:51:21:467	11:51:21:494	405044
email.txt	6M7O4H0G6R8W0F	1	11:53:44:547	11:53:44:562	7213
anbcd.doc	6X8Y7T9L7S5M7V	0	11:56:22:340	11:56:22:358	26624
d2.txt	3X5W2X6L7N8R7I	0	11:57:1:408	11:57:1:416	4392
MyApplet.java	0C2M3S8A5P	0	11:57:54:637	11:57:54:652	180
wd2.html	3K5U0L4I9E9A3V	0	11:58:36:901	11:58:36:909	2348

Table 3: Data Status at the Client

File	Bit Value	Key	client	size	open
MyApplet.java	0	0C2M3S8A5P	u1	180	no
ab15.txt	1	0V5R7O6N7E3L7E	u1	29789	yes
pdf1.pdf	1	1F2B9J1X9Y	u1	405044	yes
sd.rtf	1	1Y3Z4K8S4U7P1J	u1	1008	yes
wd2.html	0	3K5U0L4I9E9A3V	u1	2348	no
d2.txt	0	3X5W2X6L7N8R7I	u1	4392	no
pdf2.pdf	1	4Z3C2O4K5W4O	u1	20859	yes
wd17.html	1	5C5P0C1D7N3J8S	u1	241889	yes

email.txt	1	6M7O4H0G6R8W0F	u1	7213	yes
anbcd.doc	0	6X8Y7T9L7S5M7V	u1	26624	no

Table 4: Data Status after attack

File	size	Attack time	Server time	Key	client	status
email.txt	7203	9:51:45:442	9:51:14:799	2Q1H2L9T8C	u1	yes
anbcd.doc	26138	11:56:37:238	11:56:22:340	6X8Y7T9L7S5M7V	u1	yes
d2.txt	4366	11:57:19:656	11:57:1:408	3X5W2X6L7N8R7I	u1	yes
MyApplet.java	179	11:58:11:228	11:57:54:637	0C2M3S8A5P	u1	yes
wd2.html	2334	11:58:55:56	11:58:36:901	3K5U0L4I9E9A3V	u1	yes

Table 5: Overall Data Send Status

File	size	Key	Process time
ab15.txt	29789	1	9.84
email.txt	7213	0	6.655
anbcd.doc	26624	0	7.869
pdf1.pdf	405044	1	6.11
pdf2.pdf	20859	1	7.053
pdf2.pdf	20859	1	8.607
pdf2.pdf	20859	1	8.252
pdf2.pdf	20859	1	12.291
wd2.html	2348	0	7.981
d2.txt	4392	0	8.0
wd17.html	241889	1	8.174
sd.rtf	1008	1	8.072
sd.rtf	1008	1	7.155
pdf1.pdf	405044	1	9.821
MyApplet.java	180	0	8.119
email.txt	7213	1	6.305
anbcd.doc	26624	0	11.012
d2.txt	4392	0	8.594
MyApplet.java	180	0	9.479
wd2.html	2348	0	11.452

Table 6: Data Exist Status

File	up_date	up_time	del_date	del_time
d2.txt	16-1-2015	10:5:17:590	16-1-2015	10:35:47:756
d2.txt	16-1-2015	10:35:35:947	16-1-2015	10:35:47:756
d2.txt	16-1-2015	10:44:5:279		
Appendix(A).pdf	16-1-2015	10:45:11:387	16-1-2015	10:45:34:643
sd.rtf	16-1-2015	10:46:35:137		

Table 7: Security Status

File	encry_time	decry_time	edate
wd17.html	139	118	16-1-2015
sd.rtf	28	105	16-1-2015
sd.rtf	34	105	16-1-2015
email.txt	179	19	17-1-2015
pdf1.pdf	17	252	17-1-2015
MyApplet.java	19	58	17-1-2015
email.txt	24	19	17-1-2015

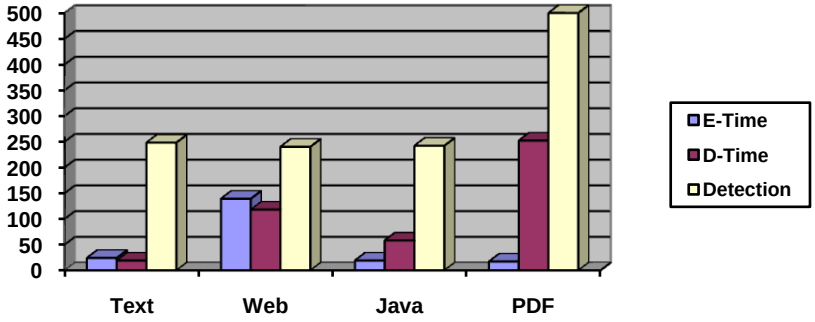

Figure 1: Comparison Chart

5. CONCLUSIONS

In this dissertation we proposed an automated server client communication system which provides better security along with timely alert in terms of data communication. In our system if data request is arrived from client to server, server automates the process if client is authorized. Then encryption and partition algorithms on requested data are applied by the server before sending the requested data. Hidden numeric adder concept is induced for preventing any unauthorized access in the communication area which also ensures that the client must beware of the content sniffed data. The proposed framework reduces the process time overhead at the server. Alongside the methodology time the reaction time is diminished to a much more noteworthy degree as contrasted with the past methodologies it additionally dynamism the presence of information document. The empowering results will outline the effect of our methodology. In future we can apply our approach on different heterogeneous files like images, video etc[32][33][34]. We can also enhance the security mechanism (Hybrid encryption algorithm).

6. REFERENCES

[1] Anton Barua, Hossain Shahriar, and Mohammad Zulkernine, "Server Side Detection of Content Sniffing Attacks", 2011 22nd IEEE International Symposium on Software Reliability Engineering.

[2] Richard Sharp and David Scott," Abstracting Application Level Web Security," In Proceedings of the 11th ACM International World Wide Web Conference (WWW 2002), May 7-11, 2002.

[3] Peter wurzinger, Christian Platzer, Christian Ludl, and Christopher Kruegel," SWAP: Mitigating XSS Attacks using a Reverse Proxy," In proceedings of the 2009 ICSE Workshop on Software Engineering for secure systems,pp.33-39,2009.

[4] Syed Imran Ahmed Qadri, Prof. Kiran Pandey, "Tag Based Client Side Detection of Content Sniffing Attacks with File Encryption and File Splitter Technique", International Journal of Advanced Computer Research (IJACR), Volume-2, Number-3, Issue-5, September-2012.

[5] Animesh Dubey, Ravindra Gupta, Gajendra Singh Chandel," An Efficient Partition Technique to reduce the Attack Detection Time with Web based Text and PDF files", International Journal of Advanced Computer Research (IJACR),Volume-3 Number-1 Issue-9 March-2013.

[6] Gupta, Saket. "Secure and Automated Communication in Client and Server Environment." International Journal of Advanced Computer Research (IJACR), Volume-3, Number-4, Issue-13, December-2013.

[7] Engin Kirda, Nenad Jovanovic, Christopher Kruegel and Giovanni Vigna,"Client-Side Cross-Site Scripting Protection," ScienceDirect Trans.computer and security ,pp.184-197,2009.

[8] Nao Ikemiya and Noriko Hanakawa, "A New Web Browser Including A Transferable Function to Ajax Codes", In Proceedings of 21st IEEE/ACM International Conference on Automated Software Engineering (ASE '06), Tokyo, Japan, pp. 351-352, September 2006.

[9] Kiezun, Adam, Vijay Ganesh, Philip J. Guo, Pieter Hooimeijer, and Michael D. Ernst. "HAMPI: a solver for string constraints." In Proceedings of the eighteenth international symposium on Software testing and analysis, pp. 105-116. ACM, 2009.

[10] Savitha Raj.S, Merlin Sharmila.A, Poorinima Beneta.P, " Hybrid Cryptographic Processor for Secure Communication Using FPGA", International Journal of

Advanced Computer Research (IJACR), Volume-3, Issue-13, December-2013 ,pp.319-324..

[11] Tateishi, Takaaki, Marco Pistoia, and Omer Tripp. "Path- and index-sensitive string analysis based on monadic second-order logic." ACM Transactions on Software Engineering and Methodology (TOSEM) 22, no. 4 (2013): 33.

[12] Yu, Fang, Muath Alkhalaf, and Tevfik Bultan. "Patching vulnerabilities with sanitization synthesis." In Proceedings of the 33rd International Conference on Software Engineering, pp. 251-260. ACM, 2011.

[13] Yu, Fang, Tevfik Bultan, and Ben Hardekopf. "String abstractions for string verification." In Model Checking Software, pp. 20-37. Springer Berlin Heidelberg, 2011.

[14] Zheng, Yunhui, and Xiangyu Zhang. "Static detection of resource contention problems in server-side scripts." In Proceedings of the 34th International Conference on Software Engineering, pp. 584-594. IEEE Press, 2012.

[15] Prabal Banerjee, Purnendu Mukherjee, Asoke Nath, " Modified Multi Way Feedback Encryption Standard (MWFES) Ver-I ", International Journal of Advanced Computer Research (IJACR), Volume-3, Issue-13, December-2013, pp.344-351.

[16] Manju Kaushik, Gazal Ojha, "Attack Penetration System for SQL Injection", International Journal of Advanced Computer Research (IJACR), Volume-4, Issue-15, June-2014, pp.724-732.

[17] Saxena, Prateek, Devdatta Akhawe, Steve Hanna, Feng Mao, Stephen McCamant, and Dawn Song. "A symbolic execution framework for javascript." In Security and Privacy (SP), 2010 IEEE Symposium on, pp. 513-528. IEEE, 2010.

[18] Urmi Chhajed, Ajay Kumar, "Detecting Cross-Site Scripting Vulnerability and performance comparison using C-Time and E-Time", International Journal of Advanced Computer Research (IJACR), Volume-4, Issue-15, June-2014, pp.733-740.

[19] Bjørner, Nikolaj, Nikolai Tillmann, and Andrei Voronkov. "Path feasibility analysis for string-manipulating programs." In Tools and Algorithms for the Construction and Analysis of Systems, pp. 307-321. Springer Berlin Heidelberg, 2009.

[20] Ofuonye, E.; Miller, J., "Resolving JavaScript Vulnerabilities in the Browser Runtime," Software Reliability Engineering, 2008. ISSRE 2008. 19th International Symposium on, vol., no., pp.57, 66, 10-14 Nov. 2008.

[21] Fadlullah, Z.M.; Taleb, T.; Vasilakos, A.V.; Guizani, M.; Kato, N., "DTRAB: Combating Against Attacks on Encrypted Protocols Through Traffic-Feature Analysis," Networking, IEEE/ACM Transactions on, vol.18, no.4, pp.1234,1247, Aug. 2010.

[22] Mathur, S.; Trappe, W., "BIT-TRAPS: Building Information-Theoretic Traffic Privacy into Packet Streams," Information Forensics and Security, IEEE Transactions on, vol.6, no.3, pp.752, 762, Sept. 2011.

[23] Qurashi, U.S.; Anwar, Z., "AJAX based attacks: Exploiting Web 2.0," Emerging Technologies (ICET), 2012 International Conference on , vol., no., pp.1,6, 8-9 Oct. 2012.

[24] Beekhof, F.; Voloshynovskiy, S.; Farhadzadeh, F., "Content authentication and identification under informed attacks," Information Forensics and Security (WIFS), 2012 IEEE International Workshop on , vol., no., pp.133,138, 2-5 Dec. 2012.

[25] Prem, M.V.; Swamynathan, S., "Securing mobile agent and its platform from passive attack of malicious mobile agents," Advances in Engineering, Science and Management (ICAESM), 2012 International Conference on , pp.605,609, 30-31 March 2012.

[26] Jagnere, P., "Vulnerabilities in social networking sites," Parallel Distributed and Grid Computing (PDGC), 2012 2nd IEEE International Conference on, pp.463, 468, 6-8 Dec. 2012.

[27] Nagarjun, P.M.D.; Kumar, V.A.; Kumar, C.A.; Ravi, A., "Simulation and analysis of RTS/CTS DoS attack variants in 802.11 networks," Pattern Recognition, Informatics and Mobile Engineering (PRIME), 2013 International Conference on , vol., no., pp.258,263, 21-22 Feb. 2013

[28] Seungoh Choi, Kwangsoo Kim, Seongmin Kim, and Byeong-hee Roh," Threat of DoS by Interest Flooding Attack in Content-Centric Networking" IEEE 2013.

[29] Ruse, M.E.; Basu, S., "Detecting Cross-Site Scripting Vulnerability Using Concolic Testing," Information Technology: New Generations (ITNG), 2013 Tenth International Conference on , vol., no., pp.633,638, 15-17 April 2013.

[30] Zheng, Yunhui, and Xiangyu Zhang. "Path sensitive static analysis of web applications for remote code execution vulnerability detection." In Proceedings of the 2013 International Conference on Software Engineering, pp. 652-661. IEEE Press, 2013.

[31] Rivest, R.L., Robshaw, M.J.B., Sidney, R., & Yin, Y.L (1998a). "The RC6 Block Cipher." URL: ftp://ftp.rsasecurity.com/pub/rsalabs/rc6/rc6v11.pdf

[32] Namrata Shukla, "Data Mining based Result Analysis of Document Fraud Detection", International Journal of Advanced Technology and Engineering Exploration (IJATEE), Volume-1, Issue-1, December-2014, pp.21-25.

[33] Bhupendra Singh Thakur, Sapna Chaudhary, " Content Sniffing Attack Detection in Client and Server Side: A Survey ", International Journal of Advanced Computer Research (IJACR), Volume-3, Issue-10, June-2013, pp.7-10.

[34] Subrata Kumar Das, Md. Alam Hossain, Md. Arifuzzaman Sardar, Ramen Kumar Biswas, Prolath Dev Nath, " Performance Analysis of Client Side Encryption Tools " , International Journal of Advanced Computer Research (IJACR), Volume-4, Issue-16, September-2014 ,pp.888-897.