iJRASET
International Journal For Research in Applied Science and Engineering Technology

INTERNATIONAL JOURNAL FOR RESEARCH

IN APPLIED SCIENCE & ENGINEERING TECHNOLOGY

Volume: 9 **Issue:** V **Month of publication:** May 2021

DOI: https://doi.org/10.22214/ijraset.2021.34604

www.ijraset.com

Call: 08813907089 | E-mail ID: ijraset@gmail.com

International Journal for Research in Applied Science & Engineering Technology (IJRASET)
ISSN: 2321-9653; IC Value: 45.98; SJ Impact Factor: 7.429
Volume 9 Issue V May 2021- Available at www.ijraset.com

Vulnerability Assesment in Web based Applications

Sri G.V. Pradeep Kumar[1], Vamshi Krishna Motru[2], Ch Mvn Sai Teja Prashanth[3]
[1]*Assistant Professor,* [2,3]*UG Student, Department of Electronics and Communication Engineering, Chaitanya Bharathi Institute of Technology, Gandipet, Hyderabad, India.*

Abstract: ***This Project depicts an in-depth technical approach to perform manual Vulnerability Assessment and infiltration test in web applications for testing the integrity and security of the application and furthermore serves as a manual for test in SANS Top 25 security vulnerabilities. The project is more centered around giving definite information about manual web application penetration testing techniques to get them from malignant black hat hackers. A victim's website can be used for criminal activities, while illegally using the website's bandwidth and making its owner liable for these unlawful acts. While developing the websites, many times developers/site owners forget to remove sensitive data from the website which is not supposed to be exposed to public users. Such data consists of untested vulnerable forms, database backup, and site backup in compressed format. A hacker tries to search for such kind of data and tries to collect important information from it like login detail from that data. So, we will be analyzing all the ways to find such vulnerabilities and also developing the patch to avoid such attacks and prevent them from occurring.***

Keywords: Vulnerability Assessment, Penetration Testing, Hacker, DataBase Back up, Sensitive Data, Patch.

I. INTRODUCTION

Cyber Security refers to the advances, cycles, and practices intended to ensure networks, gadgets, applications, and information from any sort of cyber-attacks. Cybersecurity may likewise be known as data innovation security. It is tied in with shielding devices and organization from unapproved access or adjustment. The Internet isn't just the central wellspring of data; however, it is additionally a medium through which individuals work together. Today, individuals utilize the web to promote and sell items in different structures, speak with their retailers, and perform monetary exchanges. Because of this, programmers and cybercriminals utilize the web as an instrument to spread malware and complete cyber-attacks. Along these lines, the job of Cyber Security assumes a significant part in Hospitality, Defence, Government areas, Private area, they basically assist us with ensuring our information being penetrated and give all necessary organization security. Cybersecurity is a subset of Information Technology It is basically used to ensure both programming and equipment parts. Along these lines, in the coming years, this innovation assumes an essential part. There are a few things identified with Cybersecurity like Physical security, Network security, Application security, Information security, these are to be ensured in the right manner for any fruitful working of any organization. Moral programmers are the systems administration specialists who attempt to infiltrate frameworks and discover weaknesses. Moral programmers follow a few Ethics and follow procedures to clear the Cases. The present coordinated cybercrimes out of sight solitary programmers of the past now enormous coordinated wrongdoing rings work like new businesses and regularly utilize exceptionally prepared engineers who are continually improving on the web attacks. "Cybersecurity alludes to a bunch of strategies used to ensure the uprightness of organizations, projects, and information from attack, harm or unapproved access." From a registering perspective, security contains cybersecurity and actual security, both are utilized by endeavors to secure against unapproved admittance to server farms and other modernized frameworks. Data security, which is intended to keep up the privacy, honesty, and accessibility of information, is a subset of cybersecurity. The utilization of cybersecurity can help forestall cyber-attacks, information breaks, and data fraud and can help in risk the executives.

II. LITERATURE REVIEW

Nowadays, cybersecurity has been a day-by-day issue that can be found anyplace, from the news that reports spam, frauds, scams, and wholesale fraud, to scholarly articles that talk about cyber fighting, cyber surveillance, and cyber defense. eventually, it stays a confounded assignment to move toward cybersecurity as just a basic issue of 'network security or 'individual security' as it associates with a bigger issue of "the state," "society," "the country," and "the economy". A tool purposed named SQLIVDT is designed for efficient SQLI vulnerability detection. The main goal of that tool is to generate test inputs & assess test results. Web application vulnerabilities allow attackers to perform malicious actions from unauthorized ACCOUNT ACCESS. In the last decade, web application vulnerabilities are growing. The black box approach is based on the simulation of SQL attacks against web applications.[1]. Classification of software security approaches used to develop secure software in various phases of software development life cycle.

©IJRASET: All Rights are Reserved

Static analysis approaches are able to find out the cause of a security problem and can find errors. Error finding not only reduces the cost of error but also a quick feedback cycle improves the coding approach. The static analysis approach suffers from false-positive and false-negative results. Dependence on the web application is increasing very rapidly in recent time and create a problem and for other purposes. Sql Injection and XSS are the most dangerous security vulnerability exploited in web applications i.e., eBay, Google, Fb, etc. Most developers repeat the same programming mistake in their code because they do not follow security guidelines.[2]. A set of hybrids (static dynamic) code credits that describe input approval and information sterilization code designs and are required to be huge pointers of web application vulnerabilities. Since static and dynamic program examinations supplement one another, the two methods are utilized to extricate the proposed credits in an exact and versatile manner. Therefore, building predictors using machine learners trained with the information provided by both static and dynamic analyses and available vulnerability information, achieve good accuracy while meeting scalability requirements

III. VULNERABILITY METHODOLOGIES

A vulnerability evaluation is an efficient survey of security shortcomings in a data framework. It assesses if the framework is helpless to any known weaknesses, allots seriousness levels to those weaknesses, and suggests remediation or relief, if and at whatever point required.

A. SQL Injection

SQL injection is a web security vulnerability that permits an attacker to adjust the SQL inquiries made to the database as shown in Figure 1.1 and the attacker can able to send the crafted queries and confuse the database and get access to the Database and get access into it. This can be utilized to recover some sensitive data, similar to database structure, Tables, sections, and their fundamental data SQL injection is a technique for the attack where an attacker can misuse weak code and the kind of data an application will acknowledge and can be abused in any application boundary that impacts a database inquiry. Models incorporate boundaries inside the actual URL, post data, or treat esteems. On the off chance, fruitful SQL Injection can give an attacker admittance to backend database substance, the capacity to distantly execute framework orders, or in certain conditions the way to assume responsibility for the worker facilitating the database. Proposals incorporate utilizing a layered way to deal with security that incorporates using defined questions while tolerating client input, guaranteeing that solitary expected data is acknowledged by an application, and solidifying the database worker to keep data from being gotten to improperly. Generally, SQL Injection is an attack upon the web application, not simply the web worker or the working framework itself. As the name suggests, SQL Injection is the demonstration of adding a sudden SQL order to an inquiry, consequently controlling the database in manners accidental by the database director or engineer. When effective, data can be separated, altered, embedded, or erased from database workers that are utilized by weak web applications. In specific conditions, SQL Injection can be used to assume total responsibility for a framework.

Fig 1.1SQL Injection

1) Patch: Without adequate removal or quoting of SQL syntax structure in client controllable information sources, the created SQL query can make those sources of info be deciphered as SQL rather than conventional client information. This can be utilized to change query logic to bypass security checks, or to embed extra queries that alter the back-end data set, perhaps including the execution of system commands' injection has become a typical issue with information base driven sites. The flaw is effectively-recognized, and effortlessly misused, and all things considered, any site or software package with even a minimal user base is likely to be subject to an attempted attack of this kind. This flaw depends on the fact that SQL makes no real distinction between the control and data planes. And below is the patch for this vulnerability.

```
connection. Query ("SELECT * FROM bank accounts WHERE dob =? AND bank account=? “,
[ req.body.dob,req. body. account number], // Here the input data is verified with DOB, account number. if matched function (error, results) {});
 // gives the data, else gives an error in the result
```

©IJRASET: All Rights are Reserved

B. File Upload

Permitting file uploads by end-clients, particularly whenever managed without a full comprehension of the risks related to it, is much the same as opening the conduits for worker bargain. Normally, notwithstanding the security concerns encompassing the capacity for end clients to upload files, it is an undeniably regular prerequisite in current web applications. File uploads convey a critical risk that very few know about, or how to relieve against mishandles. Most noticeably terrible still, a few web applications contain dizzy, unlimited file upload instruments. A basic file upload structure regularly comprises an HTML structure that is introduced to the customer and a worker-side content that measures the file being uploaded. Which permits the attacker to upload or move files of hazardous sorts that can be consequently prepared inside the item's current circumstance. File upload usefulness is normally connected with various weaknesses, including:

1) File path traversal
2) Persistent cross-site scripting
3) Placing of other client-executable code into the domain
4) Transmission of viruses and other malware
5) Denial of service

a) Patch: The "unrestricted file upload" term is used in vulnerability databases and elsewhere, but it is insufficiently precise. The phrase could be interpreted as the lack of restrictions on the size or number of uploaded files, which is a resource consumption issue. And below is the patch for this vulnerability.

```
// Only allowing images
allowedMimes = array (
'jpg|jpeg|jpe' => 'image/jpeg',
'gif'          => 'image/gif',
'png'          => 'image/png',
);
$fileInfo = wp_check_filetype(basename($_FILES['wpshop_file'] ['name']), $allowedMimes);
fileInfo = wp_check_filetype(basename($_FILES['wpshop_file'] ['name']));
if (! empty($fileInfo['ext']))
{// This file is valid}
else
{// Invalid file}
```

C. CSRF

Cross-site request forgery (otherwise called CSRF) is a web security vulnerability that permits an attacker to incite clients to perform activities that they don't plan to perform. It permits an attacker to somewhat evade a similar beginning approach, which is intended to keep various websites from meddling with one another's is a sort of attack which deceives the casualty to do the malignant assignment on a casualty validated web application for attacker's inclinations. The level of the attack depends fair and square of advantages that the casualty had. Since attacker will utilize the confirmation that has acquired in the current meeting to do the malevolent errand. This is the motivation behind why this attack named as Session Riding as well. CSRF attack will misuse the idea that if the client is verified every one of the requests that come from that client should be begun by the client. The attacker will misuse this idea by recognizing the meeting treat of the meeting and utilize that to send his own payload to run on the application. CSRF will possibly work if the potential casualty is verified. Utilizing a CSRF attack an attacker can sidestep the validation interaction to enter a web application as shown in Figure 1.2. At the point when a casualty with extra advantages performs activities that are not open to everybody, which is when CSRF attacks are used.

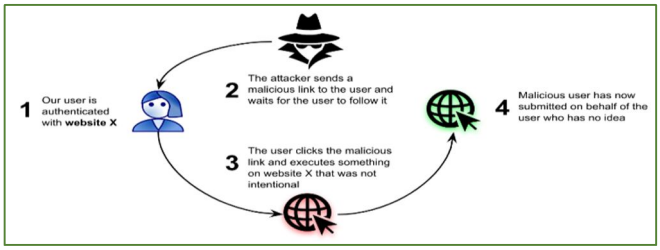


Fig 1.2 CSRF attack

©IJRASET: All Rights are Reserved

1) Patch: In this CSRF attack, the anti-CSRF tokens are added for every login attempt, and for each user account, the tokens are randomly added for every user distinctly. Below is the code for this vulnerability.

```
$_SESSION['token'] = bin2hex (random bytes (24));
if (hash equals($_SESSION['token'’],
$_POST['token']))
//Random token is generated with 24-bit length
{// Action if token is valid}
Else
{ // Action if token is invalid}
<form action="/transfer.do" method="post">
<input type="hidden" name="CSRF Token
“value="OWY4NmQwODE4ODRjN2Q2NTlhMmZlYWEwYzU1YWQwMTVhM2JmNGYxYjJiMGI4MjJjZDE1ZDZMGYwMG
EwOA== ">
</form>
```

D. Cross-Site Scripting

A Cross-Site Scripting (XSS) vulnerability was recognized in the web application. In the Database shown in Figure 1.3 cross-Site Scripting happens when powerfully created web pages show client input, for Model, login data, that isn't as expected approved, permitting an attacker to install pernicious contents into the produced page and afterward execute the content on the machine of any client that sees the website. In this model, the web application was helpless against a programmed payload, which means the client basically needs to visit a page to make the noxious contents execute. In the event that effective, Cross-Site Scripting weaknesses can be misused to control or take serve, make demands that can be confused with those of a substantial client, bargain classified data, or execute pernicious code on end client frameworks. Suggestions incorporate carrying out secure programming procedures that guarantee legitimate filtration of client-provided information and encoding all client-provided information to forestall embedded contents being shipped off end clients in an arrangement that can be executed. It is quite possibly the most widely recognized application-layer web attack. XSS weaknesses target scripts installed on a page that is executed on the customer side (in the client's web program) instead of on the worker side. XSS in itself is a danger that is achieved by the web security shortcomings of customer side prearranging dialects, like HTML and JavaScript. The idea of XSS is to control the customer-side contents of a web application to execute in the way wanted by the noxious client. Such control can insert content in a page that can be executed each time the page is stacked, or at whatever point a related occasion is performed.

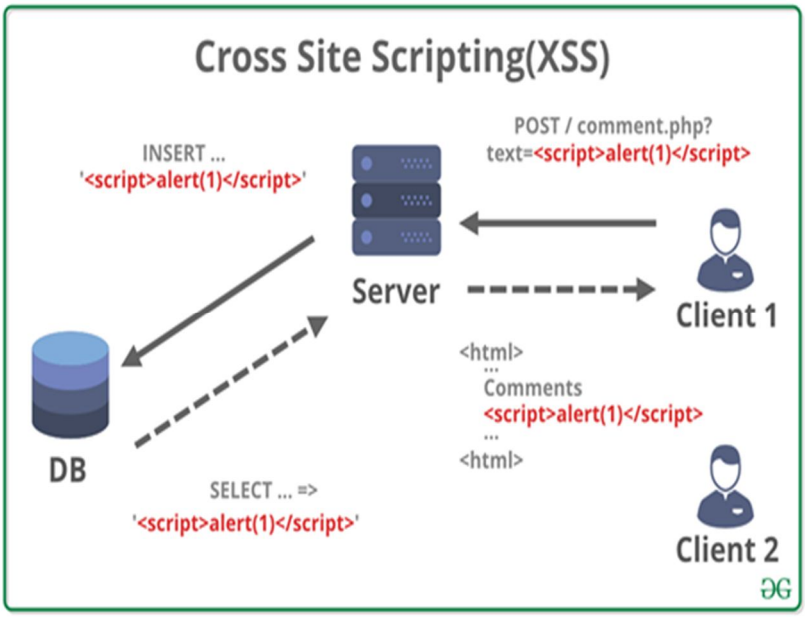


Fig 1.3 Cross-site scripting

©IJRASET: All Rights are Reserved

1) *Patch:* The vulnerability doesn't kill or mistakenly kills client controllable contribution before it is set in yield that is utilized as a site page that is served to different clients. And below is the patch for this vulnerability.

```
<script>
function arch{}
{
   txtsrcval = document.frmsearch.txtsrchval.value;
   var find;
   find= /script/;
   var patr. txtsrchval.test(find);
   if (patr == true)
   {
      alert("error");
      document. frmsearch.txtsrchhval. focus ();
      return false;
   }
   if (textsrchival !=“”}
   {
      var srchid=document. frmsearch.txtsrchval.value;
      document. fresearch. action ="search-results.php?txtsrchval= “  document.frmsearch.submit ();
   }
}
</script>
```

E. Rate Limit

A Rate Limiting is a type of attack in which sending multiple requests to the server is shown in Figure 1.4. It is the strategy for limiting the network traffic. Restricted resources incorporate memory, file system storage, data set association pool sections, and CPU. In the event that an attacker can trigger the distribution of these restricted resources, yet the number or size of the resources isn't controlled, at that point, the attacker could cause a denial of service that consumes all available resources. This would prevent valid users from accessing the software, and it could potentially have an impact on the surrounding environment. For example, a memory exhaustion attack against an application could slow down the application as well as its host operating system.

1) *Patch:* ReCAPTCHA is a service from Google that helps protect websites from spam and abuse. A “CAPTCHA” is a Turing test to tell humans and bots apart. It is easy for humans to solve, but hard for “bots” and other malicious software to Figure out. By adding reCAPTCHA to a site, you can block automated software while helping your welcome users to enter with ease. To use Google’s reCAPTCHA service, follow these steps:

a) Log into Google reCAPTCHA account and register a new site to use reCAPTCHA on.
b) Select reCAPTCHA v3.
c) Once registered a Site Key and a Secret Key is provided.
d) Next, in the website account open the Site Management page and edit the domain to add Google reCAPTCHA to.
e) In the Site edit screen, add the Site Key and Secret Key were provided by Google in Step 2 to the corresponding fields.

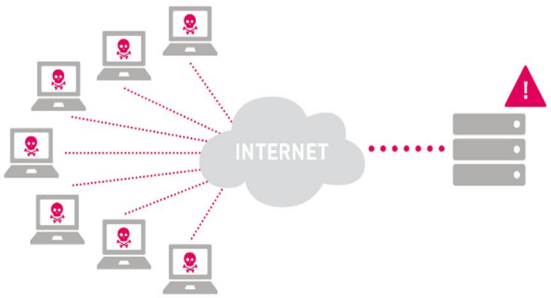


Fig 1.4 No Rate Limit

©IJRASET: All Rights are Reserved

ISSN: 2321-9653; IC Value: 45.98; SJ Impact Factor: 7.429
Volume 9 Issue V May 2021- Available at www.ijraset.com

IV. CONCLUSION

The paper gives a similar investigation between the OWASP top 10 and SANS 25 where the SANS25 vulnerabilities are been thought about and Vulnerability assessments are done on five vulnerabilities. This paper gives the vulnerable parameters in web-based applications which haven't been considered as contrasted and OWASP TOP 10 vulnerabilities utilizing static examination programming testing. It clarified how Vulnerability Assessment and Penetration Testing can be utilized as compelling cyber defense innovation. This paper gives a total outline of Vulnerability Assessment and Penetration Testing, and it is utilized as a cyber-defense technology. To get more effective and precise outcomes dissected all the paraments are classified and planned with SANS 25 vulnerabilities. So, to initiate the Penetration Testing method, it needs a whole vulnerability assessment scan to be done, to work out any vulnerabilities present within the system. Once the vulnerabilities are known, the tester further moves on to use them with vulnerability Assessment, a tester will solely get to understand the potential vulnerabilities and leave them fallow up to the current purpose. it's vital to understand the distinction, significance, purposes, and outcome of every check. Lack of data and coaching in context to each test may create a larger security risk. Organizations ought to ask business specialists to research and perceive that assessment or test works for them to fortify the application security pose. The main conclusion of VAPT is that to create the software and networks safe and vulnerable-free. Also, to scale back the cybercrimes by educating or making awareness within the individuals on a way to the aggressor will steal the sensitive info and the way to stay safely within the internet primarily based Applications. This paper additionally provides basic information on Vulnerabilities and the way to stop them by providing relevant proof of ideas in planet applications to the developers.

REFERENCES

[1] Zoran Djuric, (2013), “A Black-box Testing Tool for Detecting SQL Injection Vulnerabilities”, IEEE Second International Conference on Informatics and Applications, pp. 216- 221.
[2] Mukesh Kumar Gupta, M.e. Govil and Girdhari Singh, “Static Analysis Approaches to Detect SQL Injection and Cross Site Scripting Vulnerabilities in Web Applications”, IEEE International Conference, May 2014.
[3] Chandershekhar, Dr. S.c. Jain, “Analysis and Classification of SQL Injection Vulnerabilities and Attacks on Web Applications”, IEEE International Conference on Advances in Engineering & Technology Research,2014
[4] Lwin Khin Shar, Lionel C. Briand and Hee Beng Kuan Tan, “Web Application Vulnerability Prediction using Hybrid Program Analysis and Machine Learning”, IEEE
[5] 36th International Conference on Computer Software and Applications,2013.
[6] S.B. Chavan and B.B. Meshram, “Classification of Web Application Vulnerabilities”, IJESIT Vol2, issue 2, March 2013.
[7] Theodoor Scholte, William Robertson, Davide Balzarotti, EnginKirda,, “Preventing Input Validation Vulnerabilities in Web Applications through Automated Type Analysis”, IEEE 36th International Conference on Computer Software and Applications,2012.

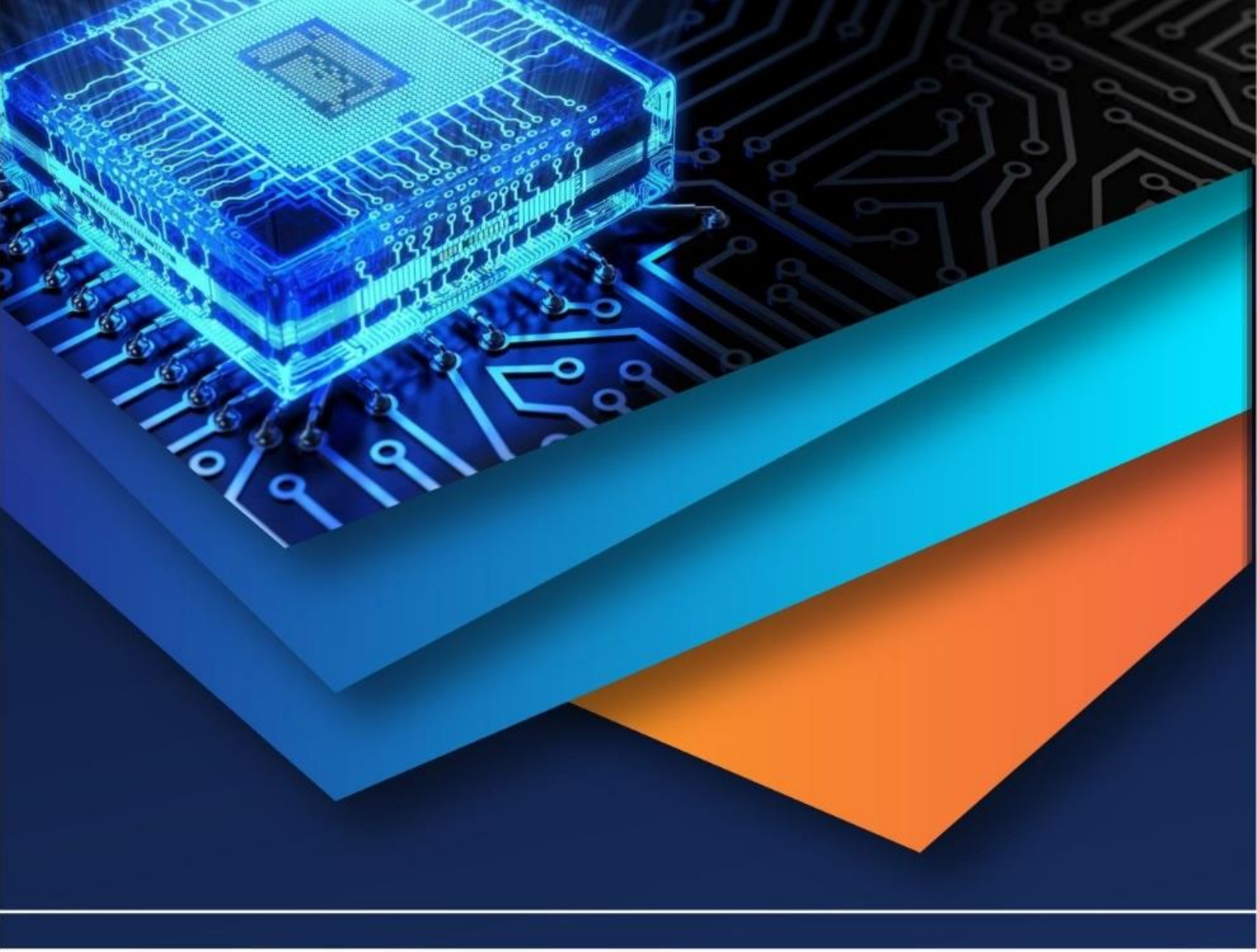
AUTHORS PROFILE

Sri G. V. Pradeep Kumar has has academic experience of 12 Years & 7 Months (Includes research experience). He is working as Assistant professor in ECE Department at Chaitanya Bharathi Institute of Technology. His educational qualification is M.E (Communication Engineering). He has published 5 research papers in International journals. His area of Speciallization is Computer networks.

Vamshi krishna Motru is a cyber security enthusiast. He has a strong knowledge of cybersecurity and proficient in risk analysis and vulnerability assessment and penetration testing. He was also featured in different MNC’c like Mastercard, Pinterest, TripAdvisor, Atlassian, DELL and Secured over 35+ companies.

Ch Mvn Sai Teja Prashanth is a ardent Security Reseacher with expertise in Cyber Security and an Hacker with Ethical mindset. He has Good experience on Web-app security, Android-app security and Vulnerability Assesment and Penestration testing. He has been featured in MNC’s like United Nations , Netflix, Indeed, Swiggy, Bigbasket and many more.

©IJRASET: All Rights are Reserved



10.22214/IJRASET

45.98

IMPACT FACTOR: 7.129

IMPACT FACTOR: 7.429

INTERNATIONAL JOURNAL FOR RESEARCH

IN APPLIED SCIENCE & ENGINEERING TECHNOLOGY

Call : 08813907089 (24*7 Support on Whatsapp)