JOURNAL OF CRITICAL REVIEWS

ISSN- 2394-5125 VOL 07, ISSUE 07, 2020

Proactively Assessing Vulnerabilities and Detecting FTP Ports Utilizing A Developed Software Suite

Hussein Abd Alwhab Ali[1], K. G. Kharade[2], Hayder Kareem Algabri[4] , R.K.Kamat[4]

Research Scholar[1,3], Assistant Professor[2], Professor[4]

husseinai322@gmail.com, kabirkharade@gmail.com [2], 7ayder.kareem@gmail.com [3]
rkk_eln@unishivaji.ac.in[3]

Department of Computer Science, Shivaji University, Kolhapur, Maharashtra, India[1,2,3]

Department of Electronics, Shivaji University, Kolhapur, Maharashtra, India[3]

Abstract: It has been decided to use new procedures to identify any possible differences that might cause harm to the network. User's security has grown more important with the introduction of modern information technology systems. In order to assess and eliminate vulnerabilities, it is necessary to have a thorough knowledge and awareness of the vulnerabilities in question. An increasing number of programs are being made available online, but how safe these goods are is a source of worry since it is tied to the security of the user who will be utilizing the application in the end. As a result, it becomes vital to identify any software program flaws that may pose a severe threat to the security of the user. In order to safeguard the social, digital environment, it is essential to conduct a thorough investigation of these flaws and vulnerabilities. Thus, there is a need to develop a framework that can identify several levels of vulnerabilities, ranging from client-side vulnerabilities through communication-side vulnerabilities and ultimately to server-side vulnerabilities. The current investigation is concerned with the construction of a vulnerability scanner. The scanner in question is specialized in scanning the FTP port before transmitting and receiving data.

Keywords: Attack graphs, FTP, Port scanning**,** Security, Vulnerability

1. Introduction: Messages must be transmitted between hosts and networks in a fast, reliable, and secure way; this is a considerable challenge in an open environment, especially when it comes to the Internet and TCP-based networks, but it is not impossible to do. Even though there are a variety of other file transfer protocols accessible in addition to the core FTP protocol, they are all slow, unreliable, and potentially dangerous in their functioning. This study has included a vulnerability evaluation process that runs in the background while sending the data. Port numbers 20 and 21 are being studied for their behaviour mechanisms since the earlier ports are being utilized for FDP. Port 21 is used to establish a connection between the two computers (or hosts), while port 20 is used to transport data between the two

ISSN- 2394-5125 VOL 07, ISSUE 07, 2020

computers (or hosts). The HTTP (HyperText Transfer Protocol) transfers hypertext and hypermedia over web information systems. A simple mail transfer protocol, such as SMTP (Simple Mail Transfer Protocol), is used for transferring e-mail between devices.

In contrast, RTP (Real-time Transport Protocol) is used to deliver audio and video over networks, and FTP (File Transfer Protocol) is used to transfer files between computer systems. It is feasible to do this by one of two approaches. For starters, manual analysis is prone to inaccuracy due to the human propensity to forgive, technology is continuously improving, and fraudulence attack strategies must be considered. As a second option, web application vulnerability scanners that are now accessible are used, which may sometimes create many false alarms owing to a high percentage of false positives, which is a concern (Al-Hakeem et al.,2013).

2. Vulnerability Scanners: An increasing number of programs are being made available online, but how safe these goods are is a source of worry since it is tied to the security of the user who will be utilizing the application in the end. As a result, it becomes vital to identify software program vulnerabilities that may pose a significant threat to the security of the user (Bairwa et al., 2014). In the field of computer security, vulnerability scanning refers to the process of looking for flaws in a computer system's defences. With the aid of the Vega vulnerability scanner, it is possible to detect and address system vulnerabilities before an attacker discovers and exploits the weakness and makes use of it. In order to patch security vulnerabilities and maintain high levels of safety for your systems, data, employees, and customers, vulnerability detection software must be used in conjunction with a proactive strategy.

In many cases, data breaches are the result of unpatched vulnerabilities; as a result, identifying and patching security gaps removes the danger vector that was previously there. Secure systems are essential for compliance with cybersecurity regulations and legislation. For example, the National Institute of Standards and Technology (NIST), the Payment Card Industry Data Security Standards Council (PCI DSS), and the HIPAA Security Standards Council encourage vulnerability screening to secure sensitive data. In addition, cybercriminals have access to vulnerability scanning tools. As a result, it is critical to conduct vulnerability scans and take corrective steps before hackers exploit any security weaknesses.

3. Challenges before Vulnerability Scanning: Several challenges arise in conducting vulnerability scanning. The majority of scans are "snapshots" rather than continuous. Because

ISSN- 2394-5125 VOL 07, ISSUE 07, 2020

your systems are constantly evolving, you should conduct scans frequently to keep up with the changes in your IT environment. In the end, a vulnerability scanning tool is only as good as its known faults and signatures database, both of which are maintained by the program's creators. Your tool will need to be updated frequently because new vulnerabilities are found all the time. Even though the scanning process may be automated, a security expert may still need to review the results, implement remediation, and monitor the situation to ensure that threats have been mitigated or eradicated. Additionally, many companies integrate vulnerability scanning with automated patch management and other technologies to reduce staff spending on administrative activities. The scan itself is just the first step in the vulnerability management lifecycle regardless of the findings. The extent to which you want your scan to be thorough will determine how extensive your scan should be. It is thus recommended that automated administration of these credentials and integration with the scanner be explored to improve the depth of scan and the security of privileged access credentials.

4. Scanning Types

a) **Passive Scanning**: To identify services, hosts, and clients, passive scanning techniques rely on the monitoring of network traffic to do so. An observation point is established on the network, which necessitates the involvement of network managers or network engineers to design these systems for the best possible outcomes. In contrast to active scanners, passive scanners may be operated continuously for extended periods without interfering with ordinary network traffic or interacting with the devices themselves since the input data for passive scanners is a direct feed of the network's traffic (Coffey et al., 2018). A passive scan analyzes network data in the background to discover endpoints and traffic patterns. Communicating directly with the endpoints generates no extra network traffic and has nearly little danger of causing necessary operations to become unresponsive or fail. On the other hand, active monitoring may take longer to gather asset data since it must wait for network traffic to or from each asset to produce an accurate profile. Passive monitoring Not all network portions are accessible in all circumstances, which may restrict the ability to passively monitor traffic throughout the complete OT environment in certain instances (Kritikos et al., 2019).

ISSN- 2394-5125 VOL 07, ISSUE 07, 2020

b) **Active Scanning:** Active scanning for system inventory information and vulnerability data is a robust technology that can provide significant advantages. Active scanning of your network, on the other hand, might cause significant difficulties. It can have a high political cost and far-reaching consequences for system uptime and dependability. Ineffective and inefficient information gathering might result if this is not done with care and precision. It is possible to do active scanning by sending test traffic into the network and querying specific endpoints. It is possible to collect basic profile information such as the device name, IP address, NetFlow or Syslog data, and more detailed configuration information. Such as the make and model of the device, firmware versions, and installed software/versions, and operating system patch levels using active monitoring. Active scanning may speed up data gathering by delivering packets directly to endpoints rather than a mediator. The danger of malfunctioning endpoints grows due to incompatible queries sending them or flooding smaller networks with considerable traffic. Furthermore, since active scanning does not often monitor the network 24 hours a day, it may miss temporary endpoints or listen-only devices on the network (Kharade et al., 2019)

5. Categories of Scanner

Various categories of a scanner are;

a) **Port Scanner**: While port scanning is a means of discovering which networks' ports are open, it is not a method of knowing which networks' ports are receiving or transmitting data. It is also a procedure that involves sending packets to specified ports on a host and evaluating the answers to determine whether or not the host is vulnerable. This scanning cannot occur without first identifying a list of active hosts and mapping the IP addresses of those hosts to their respective hostnames and domain names. This operation referred to as host finding, begins with a network scan. Scanners for ports and networks are used to detect the organization of IP addresses and hosts and the ports they use to accurately assess open or vulnerable server locations and diagnose security levels. It is possible to detect the existence of security measures such as a firewall between a server and a user's device using both network and port scanning techniques (Zeng et al., 2019).

ISSN- 2394-5125 VOL 07, ISSUE 07, 2020

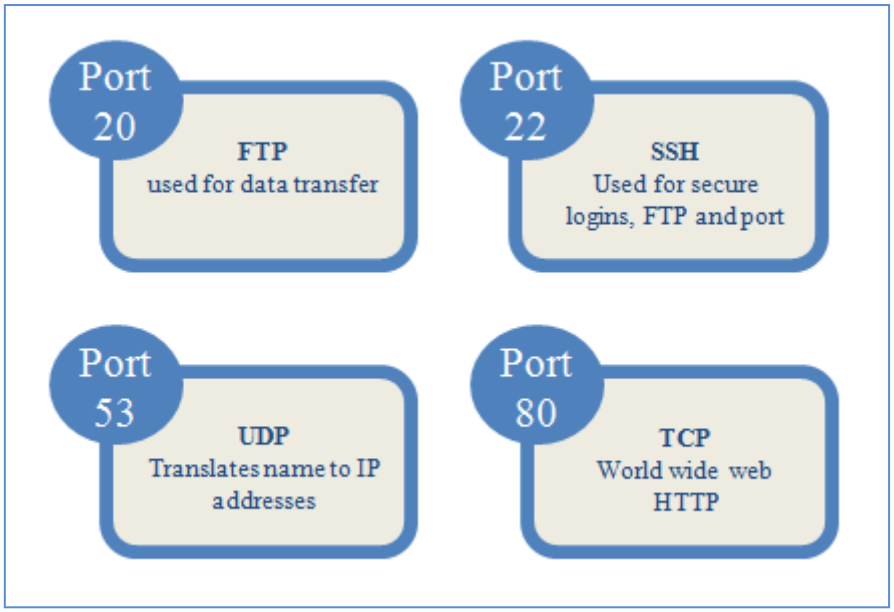

Fig. 1: Most commonly used port scanning techniques

b) **Application Scanner:** Using an application scanner, you may evaluate a program running on your network to identify and monitor any holes that might be exploited to present a security risk to the system. Web application vulnerability scanners examine websites to identify known software flaws and incorrect setups in-network or web-based software.

c) **Vulnerability Scanner:** Malicious users or hackers may exploit system flaws, putting the whole network system in danger if they access it. Vulnerability scanners are the tools that identify these vulnerabilities. In an organization's information technology architecture, vulnerability scanners are essential tools that search for and report on known flaws that have been identified. Every firm may profit from vulnerability scanners since they are easy, but they are essential security practices to implement. By analyzing possible security holes inherent in a company's environment, these scans may provide an organization with an indication of the kind of security risks they may be facing. Many businesses use numerous vulnerability scanners to guarantee that they obtain thorough coverage of every asset, resulting in a comprehensive view of their security posture. Many scanners have been produced throughout the years, each offering various choices and functions.

d) **Network-based Scanners:** Network security threats and susceptible systems on wired or wireless networks may be identified and remedied using network-based vulnerability scanners. The use of network-based scanners can aid in the discovery of

ISSN- 2394-5125 VOL 07, ISSUE 07, 2020

unknown or unauthorized devices and systems on a network. They can also aid in the determination of whether a network contains unknown perimeter points, such as unauthorized remote access servers or connections to insecure networks of business partners, that must be addressed (Latora & Marchiori, 2005).

e) **Host-based scanners:** Host-based vulnerability scanners are used to discover and identify vulnerabilities in servers, workstations, and other network hosts. They also provide deeper insight into scanned systems' configuration settings and patch history than traditional vulnerability scanners. Vulnerability assessment tools based on the host can also give insight into the possible harm that both insiders and outsiders may do after some degree of access has been allowed or taken on a system (Kharade et al., 2019).

6. Techniques for Vulnerability Scanning

a) **Static Analysis:** Static analysis is an approach that is both quick and dependable. It has been deemed to be a convenient approach to identifying security flaws. This methodology focuses on examining program structure via the use of diverse methods. It places a strong emphasis on studying the program's code to identify any defects that may be there. The static analysis employs various approaches, including lexical analysis, type inference, constraint analysis, etc. When doing lexical analysis, the emphasis is on the semantics of the program structure; the program structure is broken into modules. Each module is then compared with the loophole library to find any defects that may be present in the system. It is tied to the variable's data type rules when it comes to type inference. It checks to see whether the variables used in the program are consistent with the type to which they are related. The procedure of constraint analysis consists of two steps. It involves the creation of constraints as well as the solution of constraints. Strictly speaking, the static analysis uses tools to review program code to identify application coding flaws and back doors. It also focues on other malicious code that could allow hackers to access sensitive company data or customer information. In some circumstances, the analysis is carried out on a particular version of the source code, and in others, it is carried out on a particular form of the object code is used. During a static analysis scan of source or object code, the software's security and functionality are evaluated when the program is not

ISSN- 2394-5125 VOL 07, ISSUE 07, 2020

operating, often early in the development lifecycle. Static analysis is often carried either by a computer program or hand (Kharade et al., 2020).

b) **Attack Graph Analysis:** It is characterized as a brief description of all paths an attacker follows in a network to obtain the state it wishes. Before attaining the desired state, it may be required to harm the network, steal network packets, or get exclusive access to the network to detect what is happening. Attack graphs may imitate the different paths that attackers could take to enter a network. The attack graph enables the network administrator to examine the network's overall security and analyze and anticipate the attacker's behaviour, among other things. Even though there have been multiple research articles on attack graphs, there has been no complete examination of the accompanying analytic approaches. Finally, a comparative assessment of the methodologies and recommendations for further research are presented. We feel that our study will aid the research community in its efforts to grasp the attack graph analysis approach in a systematic and organized manner (Zeng et al., 2019).

7. FTP Server: The File Send Protocol (FTP) is a standard network protocol used to transfer files from one host to another host without a TCP-predestined network, such as the Internet. It accomplishes this purpose by using unique categorizations and data connections between the client and the server, which are constructed on a client-server paradigm. FTP is built on the client-server concept. Authentication for FTP users is accomplished by using a clear-text sign-in approach, which is often represented as a username and password combination. It is also possible for FTP users to bond surreptitiously if the server has been configured to allow it (Kharade et al., n.d.).

8. Role of FTP protocol

Our understanding of the FTP protocol includes how data may be moved via a network. The essential characteristics of the FTP protocol are as follows:

- It enables users to access distant computers via the use of applications.
- To safeguard a user's data against duplication when a file is shared across many hosts.
- To stimulate the sharing of files;
- To deliver data consistently and efficiently. FTP may be used freely by a user via a terminal, although it primarily uses software programs.

ISSN- 2394-5125 VOL 07, ISSUE 07, 2020

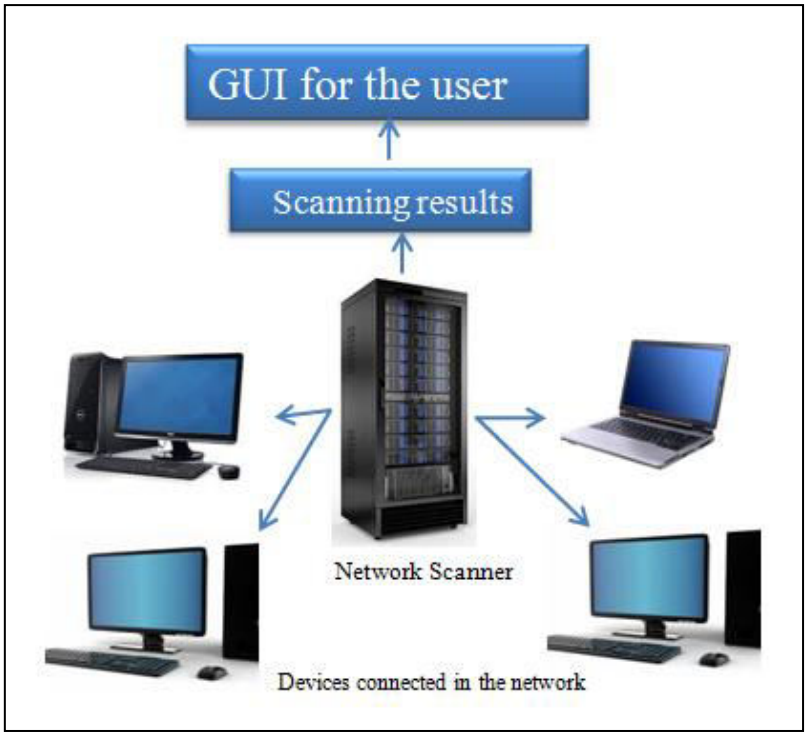

Fig. 2: Vulnerability scanning model for the FTP port

9. Working Model of the system

To identify all the ports in a network-based system

```
def vulnerable(ip,port):

    nse_ftp_scripts = ['ftp-anon',
    'ftp-bounce',
    'ftp-brute',
    'ftp-libopie',
    'ftp-proftpd-backdoor',
    'ftp-syst',
    'ftp-vsftpd-backdoor',
    'ftp-vuln-cve2010-4221',
    'tftp-enum']
```

Fig. 3: code snippet to check all the ports

To check the vulnerability of port 20 as this is an FTP port. This port is responsible for transferring files within a connected network. The mechanism of transferring files will identify the vulnerability in a network. If the vulnerability is found in the network, transferring a file will be cancelled.

ISSN- 2394-5125 VOL 07, ISSUE 07, 2020

```
if port == '20':
    for nse in nse_ftp_scripts:
        cmd = "sudo nmap -p {} -T4 --script={} {} --host-timeout 600000ms".format(port,nse,ip)
        result = os.popen(cmd).read()
        result = str(result)
        if "VULNERABLE" in result:
            cve = "CVE:CVE-"
            cmd = "sudo nmap -p {} --script={} {} | grep {}".format(port,nse,ip,cve)
            output = os.popen(cmd).read()
            if cve in output:
                cve = []
                ind = output.index('C')
                cve = output[ind:ind+17]
                cve_details.append(cve)
                ip_details.append(ip)
                port_details.append(port)
                nse_details.append(nse)
                info = Cve_info(date=_date,ip=ip,port=port,nse=nse,cve=cve)
                db.session.add(info)
            else:
                like_ip.append(ip)
                like_port.append(port)
                like_nse.append(nse)
                intel = Likely_info(date=_date,ip=ip,port=port,nse=nse)
                db.session.add(intel)
```

Fig. 4: code snippet to identify the vulnerability at port 20

The following diagram shows the vulnerability chart with GUI for the user.

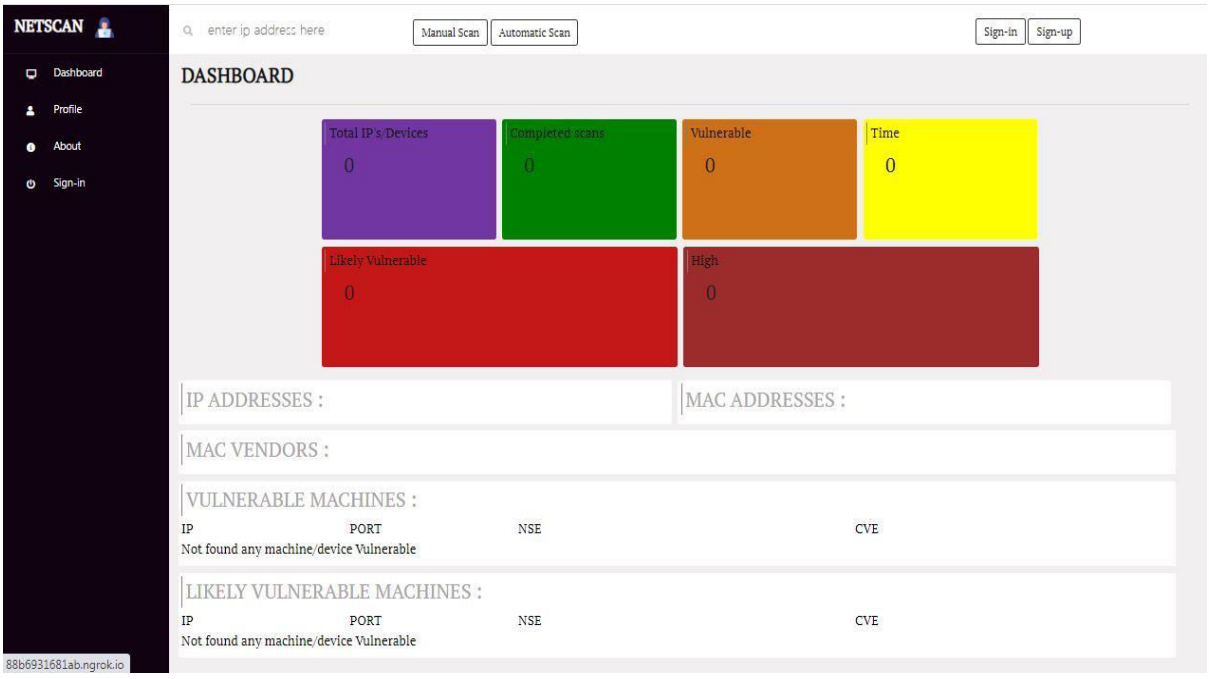

Fig. 5: Initial position of the software package

Once you begin scanning the network, it will examine all of the ports that have been specified as a part of the network scanning process. It has the capability of scanning more than 2000

ISSN- 2394-5125 VOL 07, ISSUE 07, 2020

different security scans. The system will take around 40 minutes to scan all of the security scans in their entirety. However, if you provide a restricted port scan, the time required will be insignificant. The number of computers linked to a network impacts the overall performance of the system.

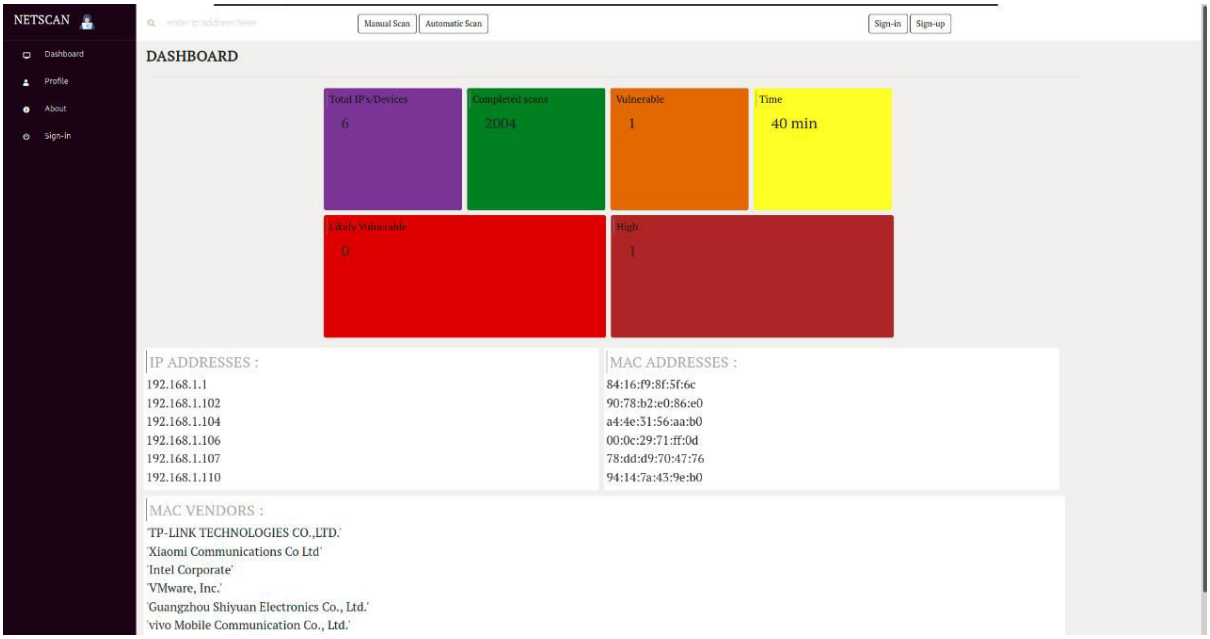

Fig 6: Scanning report for the specified port

10. Conclusion:

The newly created device can scan numerous ports simultaneously, depending on the situation. However, we primarily focused on port numbers 20 and 21 for vulnerability screening throughout our investigation. The ports mentioned above are critical in the network system's ability to transfer files. A user who wishes to transmit an unprotected file from one computer to another may expose the file to infection by various viruses throughout the process. If you transmit the infected file to another user, there is a chance that the files on that person's computer will be corrupted. In order to prevent this, the proposed model first identifies the vulnerability before distributing it to the rest of the users. Before downloading from the server, the second check entails recertifying the document.

11. References

1. Al-Hakeem, Mazin & Mohammed Zeki, Suhair & Yousif, Sarah. (2013). Development of a Speed Reliable Secure File Transfer Protocol (SRS-FTP). Al-Mansour Journal. 1.

2. Bairwa, S., Mewara, B., & Gajrani, J. (2014). Vulnerability Scanners: A Proactive Approach to Assess Web Application Security. *International Journal on Computational Science & Applications*, *4*(1), 113–124. https://doi.org/10.5121/ijcsa.2014.4111
3. Coffey, K., Smith, R., Maglaras, L., & Janicke, H. (2018). Vulnerability Analysis of Network Scanning on SCADA Systems. *Security and Communication Networks*, *2018*, 1–21. https://doi.org/10.1155/2018/3794603
4. Kharade, K. G., Kamat, R. K., & Kharade, S. K. (2019). Online Library Package to Boost the Functionality and Usability of the Existing Libraries. *International Journal on Future Revolution in Computer Science & Communication Engineering*, *5*(8), 5–7.
5. Kharade, K. G., Kharade, S. K., & Katkar, S. V. (2019). Cyber Security-A Method of Generic Authentication of Data with Ip Security. *International Journal of Information Systems*, *9*(2), 63–65.
6. Kharade, S. K., Kamat, R. K., & Kharade, K. G. (2019). Simulation of Dye Synthesized Solar Cell using Artificial Neural Network. *International Journal of Engineering and Advanced Technology (IJEAT)*, 1316–1322.
7. Kharade, K. G., Kharade, S. K., & Kumbhar, V. S. (2018). Impact of Digital India on Various Sectors. Indian Journal of Innovation in Management and Excellence In Research, 2(1), 37–40
8. Kritikos, K., Magoutis, K., Papoutsakis, M., & Ioannidis, S. (2019). A survey on vulnerability assessment tools and databases for cloud-based web applications. *Array*, *3–4*, 100011. https://doi.org/10.1016/j.array.2019.100011
9. Latora, V., & Marchiori, M. (2005). Vulnerability and protection of infrastructure networks. *Physical Review E*, *71*(1), 015103. https://doi.org/10.1103/PhysRevE.71.015103
10. Kharade, S. K., Kamat, R. K., & Kharade, K. G. (2019). Simulation of Dye Synthesized Solar Cell using Artificial Neural Network. *International Journal of Engineering and Advanced Technology (IJEAT)*, 1316–1322.
11. Kharade, K. G., Kamat, R. K., & Kharade, S. K. (2019). Online Library Package to Boost the Functionality and Usability of the Existing Libraries. *International Journal on Future Revolution in Computer Science & Communication Engineering*, *5*(8), 5–7.
12. Kharade, K. G., Kamat, R. K., Mudholkar, R. R., & Kharade, S. K. (2018). Removable Drive Blocker Application for Virus Detection. International Journal of Research Culture Society, 233-234.

ISSN- 2394-5125 VOL 07, ISSUE 07, 2020

13. Zeng, J., Wu, S., Chen, Y., Zeng, R., & Wu, C. (2019). Survey of Attack Graph Analysis Methods from the Perspective of Data and Knowledge Processing. *Security and Communication Networks*, *2019*, 1–16. https://doi.org/10.1155/2019/2031063