Vulnerability Assessment of Web Applications using Penetration Testing

Gitanjali Simran T, Sasikala D

Abstract— In recent years, utilization of web applications, web hacking exercises have grown exponentially. Organizations are confronting extremely critical difficulties in anchoring their web applications from rising cyber threats, as bargain with the assurance issues don't appear to be the right approach. Vulnerability Assessment and Penetration Testing (VAPT) methods help us find these vulnerabilities / security loopholes in our systems even before an intruder could find a way to get it. This helps avoid zero-day exploits. This paper aims to elucidate the overview of Vulnerability Assessment and Penetration Testing and introduce the most efficient open source tools used to perform these tests. This paper also presents a combined VAPT testing methodology that incorporates strengths of several existing approaches, with the goal to understand their utility and benefit the most from the tests.

Keywords—cyber security, VAPT, zero-day exploits, vulnerabilities.

I. INTRODUCTION

Threats to integrity and confidentiality of information are constantly increasing. To protect our information, we are obliged to perform security tests. Attackers are always finding new ways to exploit a system this leads to evolution of new vulnerabilities. Security testing is a process which is intended to reveal flaws in the security mechanisms and find potential vulnerabilities to check if a system is compromised. If any system is not tested for security related issues it might end up with security loopholes which may result various risk factors including loss/ leakage of information which is confidential to the organization.

Vulnerability Assessment and Penetration Screening (VAPT) are two forms of vulnerability testing. The assessments have different strengths and they are often combined to accomplish a far more complete vulnerability analysis [1]. In a nutshell, Penetration Testing and Vulnerability Assessments perform two different tasks, usually with different results, within the same area of focus.

Vulnerability assessment comprises of identifying the various loopholes or weakness in a system through which an intruder or attacker may possibly attack the system. Through vulnerability assessment we can merely only identify the risks and cannot exploit them. This is where the role of penetration testing comes in. Unlike the vulnerability assessment pen testing does not merely involve discovering the threats, we exploit the threats, or the loop holes reported by vulnerability assessment to ensure if they really exist or if those were false alarms [2]. Penetration tests find exploitable flaws and measure the severity of each exploited threat. It is always suggested to perform these two assessments together.

VAPT surveys the adequacy and inadequacy of the security courses of action of the web application to remain ensured against the rising Cyber dangers. The anticipated work builds up a flexible instrument which can discover vulnerabilities from web applications. In this way, identification of Vulnerabilities and cure of a comparable has turned out to be one among the prime issues for associations. With the developing between availability of frameworks and progression in Cyber Services, the degree of Cyber Attacks has conjointly misrepresented. In this manner as to remain resistant and for risk minimization, Vulnerability Assessment and Penetration Testing is led by the associations on customary premise.

The two sorts of security assessment are vulnerability assessment and penetration testing which can frequently be consolidated for accomplishing better powerlessness examination results.VA and PT are only two distinct errands giving diverse outcomes yet inside a similar workspace. We have Vulnerability appraisal devices for finding vulnerabilities, though no separation found between sorts of imperfections that reason harm on misuse and those that don't do as such. There are Vulnerability scanners which create caution for organizations about pre-presence of any blemishes in code and area of imperfections. Entrance tests are performed to abuse the vulnerabilities in a framework to get any method for unapproved access or probability of any malevolent movement and utilized in ID of defects presenting the danger to the application. These tests discover exploitable blemishes and measure their seriousness. These are additionally useful for demonstrating the measure of harm it could cause amid the genuine assault. In this way, joined bundle of infiltration testing and weakness appraisal instruments give a point by point perspective of existing defects alongside the hazard related with it. For a security tester to completely test a web application for security threats, he cannot stop with performing only simple web security search. A complete analysis of the system under test has to be performed and accordingly the test has to be carried forward. A repeated continuous analysis of the system is always advised [3]. To perform this just a vulnerability assessment or a penetration test is not sufficient. To perform a complete and deep security test for the web application, I propose a method which includes a series of tests which performed in the proposed order gives the most efficient results for the security test.

Revised Manuscript Received on November 15, 2019

Gitanjali Simran T, Completed her PG Bannari Amman Institute of Technology Sathyamangalam, India

Sasikala D, Department of Computer Science and Business Systems, Bannari Amman Institute of Technology, Erode,India Email: sasikalad@bitsathy.ac.in

These tests are mostly performed with the best open source tools designed for tests like port scanning, vulnerability assessments, penetration testing, Network capturing.

II. VULNERABILITY ASSESSMENT AND PENETRATION TESTING

First, It is known that web applications or websites are in general vulnerable to security attacks, be it code based or network based. These vulnerabilities give the attackers an opportunity to take control of the system and its components. This is when we call a system compromised. Not only highly used banking sites or so even a basic website written in plain simple and static html needs a detailed vulnerability assessment and penetration testing. It is important to understand the seriousness of vulnerable websites or webservers. An attacker may potentially steal sensitive data from the server or disrupt website operations or simply deface pages of the website [4]. It is crucial to realize that protecting the web application with just firewalls is not enough hence we need a periodical detailed VAPT to ensure the system is fool proof.

Vulnerabilities are framework imperfections, bugs, misconfiguration that make it defenseless against the attacks. Evaluating of these framework vulnerabilities empower us to distinguish and introduce security patches, in order to shield the framework from the danger of being harmed. VAPT strategy is directed in two noteworthy parts. The main half manages the Analysis and Discovery of existing Vulnerabilities. The second half manages the Exploitation of the distinguished arrangement of Vulnerabilities, to assess their Severity and effect over the Target framework. Vulnerability assessment is a detached methodology though penetration testing is a functioning methodology where security experts recreate assault and test the objective site and its resilience control against assaults.

III. TYPES OF VULNERABILITIES

Open Web Application Security Project (OWASP) is an organization that provides unbiased and practical, cost-effective information about computer and Internet applications. This organization has taken data and surveys from the most powerful companies across the globe and put together an OWASP Top 10 Web Application Security Risks for web applications [5]. This was put together to provide guidance to developers and security professionals on the vulnerabilities that have most risk and are commonly discovered in software applications , these are also easily exploitable.

In spite of having such clear guidelines most organizations continue to fail in protecting their systems from these common attacks or vulnerabilities which are most often simple and easy to identify and resolve. Most organizations fall prey because they have misconceptions about what a web application is. A one-time vulnerability scan or penetration assessment of a handful of business-critical apps is not effective approach to application security. There should be an approach that continuously assesses the applications an organization develops for production is effective and recommended application security.

The following threats are the OWASP Top 10 Web Application Security Risks as of 2017 OWASP survey

- Injection
- Broken Authentication and Session Management
- Sensitive Data Exposure
- XML External Entity
- Broken Access Control
- Security Misconfiguration
- Cross-Site Scripting
- Insecure deserialization
- Utilizing Components with Known Vulnerabilities
- Lacking Logging and Monitoring

IV. TEST METHODOLOGY

To perform a penetration test, a testing plan must be initially made. A plan to carry out the test is depicted in Fig.1. There are many existing methodologies and aides; a standout among the most widely recognized is the OWASP Testing Guide v4. This area will depict the areas of this arrangement, and how to test the most widely recognized security vulnerabilities. To perform an effective security test capturing most if not all the vulnerabilities of a system, a number of security tools have been used to test a web application. These tools are all specifically designed keeping the OWASP security risks as a priority to perform unique tasks and hence putting different unique tools together gives us a wide perspective.

Before a system is tested it should first be scanned for any open ports, these are the ports that will be tested as these will be the entry point for the attackers. Once the ports are scanned we need to see the intensity at which the system can withstand an attack by performing a storm test; which basically is bombarding the port with high packet rates to see how the system handles these packets. Next perform a vulnerability assessment to establish the list of the possibly vulnerable or exploitable ports or issues in the system. When we know the possible vulnerabilities, we need to exploit each of these loopholes manually to make sure the vulnerability is truly present as most of the times these tools might throw in or detect false positives which has to be checked by performing a penetration testing with the support of network capturing and monitoring tools.

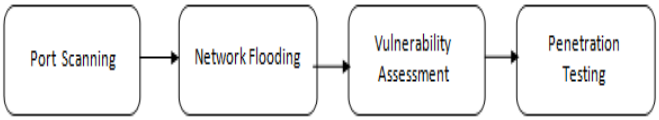


Fig.1. Test Methodology.

The following tools were used to perform the entire procedure and the results were observed for the same. The testing platform used were Kali Linux 2.0 and Ubuntu 16.4. The tests were conducted on a variety of web application, web servers and embedded systems.

A. *Nmap*

Nmap is the short form for Network Mapper. It is a free and open source utility for network discovery and security auditing. This particular tool identifies what hosts are available on the network and it also helps discovering the hosts and services (application names and versions) the devices offer.

It is by large used to find open ports and detect security risks in devices connected to the system. Namp has the capability to scan websites, servers, networks and many more for vulnerabilities. Nmap in general is port scanning / port discovery tool. It works by sending raw network packets to system ports and listens for responses from those ports and determines if those ports are open, filtered or closed. Nmap supports protocols like TCP, UDP and also ICMP. While checking for each of these protocols it uses a different packet structure. Nmap is available in Windows, Mac OS and Linux. I prefer using Nmap on Linux as it is faster and more efficient. Nmap not only has a command-line executable, it also includes an advanced GUI and results viewer for convenience. If we are confused about how to start a security, Nmap can lead you the way.

B. *Achilles*

There is likelihood that a system is prone to security flaws in Ethernet, ARP, IP, ICMP, TCP and UDP implementation which could lead to potential vulnerabilities. The Achilles Satellite Leve2 test suite uses grammar-based testing, stateful packet generation and storm tests to test for security vulnerabilities that impact the confidentiality, integrity and availability of the Device under test. Storm test determines the ability of the device to handle packets at fast rates i.e., it determines the maximum number of packets a DUT can handle at each Layer. Achilles uses automated grammar-based testing and state full packet generation to test for cyber security threats that impact the performance and integrity of a device. It is necessary to rerun a small range of test vectors to confirm the presence of anomalies. When an anomaly is reported during Achilles fuzz/ grammar test case, the subtest that was running at that time is displayed. The displayed subtest might not be the actual cause of the anomaly; the cause might be a previous subtest. Hence to reproduce the anomaly, it is necessary to narrow down the range of subtests and rerun the test, continuing in this way until the problem subtest is isolated.

C. *Nessus*

Tenable Network Security's Nessus is a popular and a very efficient vulnerability scanner. Nessus is one of those tools that are user friendly and has a great GUI. Though there are other such tools in the market, Nessus stands out because of how incredibly easy it is to use and the speed at which it works and it also gives a detailed report of the vulnerabilities which is very useful for the developers as well. Nessus has two components; Nessus Daemon and Nessus Client. Nessus Daemon (nessusd) is responsible the actual vulnerability scanning and the vulnerability tests are written in Nessus Attack Scripting language (NASL) which is optimized for custom network interaction and the Nessus client is what controls the scans run by nessusd and it produces the results of the vulnerability of the system. Nessus generally performs a port scan to identify the open ports on the target. Only after this it exploits those ports and checks for various vulnerabilities. It keeps the OWASP top 10 risks as a center of its scanning and pays a lot of attention to these vulnerabilities among others. The vulnerability checks Nessus has are called plugins. Each plugin checks for an individual vulnerability. For instance there is plugin for XSS vulnerability, SQL Injection, for issues with DHCP server and so on. Nessus has around 70,000 such plugins that are updated on a weekly basis. Nessus along with vulnerability scanning it does other basic network scans, security audits, host discoveries, compliance tests and a few other functions.

D. *Burp Suite*

Burp suite is a java based integrated platform for performing security testing of web applications[6]. Burp suite in general is a web penetration testing framework. This tool is exclusively made for web applications. There is a community edition and a professional edition both of which prove to be extremely useful. Burp suite is now used by most of the professional testers as a part of their industry standard tools. Burp allows us to perform complex and customized tasks and also write up individual plugins as per our requirements.

Burp suite is simply an intercepting proxy which helps a penetration tester to configure traffic to route through Burp. Burp suite performs in a way similar to man in the middle attack. It is placed between the web client and server, so this captures every request and response from and to the client and the server. Burp has an option called the intercept, this helps the tester to pause the traffic and manipulate the data and test how it affects the data.

The normal flow to test a website is to initially set the burp as the proxy and specify the target scope, then we can either manually crawl to each page on the web application or let Burp do it using its spidering and crawling options. Once Burp has visited each link then we can both actively and passively scan the website for vulnerabilities. Meanwhile it also has other options like the repeater, comparer and decoder which helps the tester to replay the requests or modify them and check the behaviour of the website to these requests. In addition to these it has an intruder option which allows us to perform customized brute force attacks and a sequencer which is mostly used to perform fuzzing find of operations, which is it sends the web page random data load to see how the page handles it and also to find unknown vulnerabilities.

E. *Wireshark*

Wireshark is an open source network analyzing tool to capture the traffic in a network and it also analyses the packets by categorizing them. This kind of tools can be called network sniffers as well. Wireshark was earlier known as Ethereal [7]. It is used to analyze the details of the network traffic at all levels of the OSI model. It can trace packets from the connection level to the bit level.

Wireshark uses pcap to capture packets, a .pcap file can give us the information like the packet transmit time, the source and destination IPs, the protocol involved including the header data of the protocol. These details are extremely useful to evaluate a security breach event and to troubleshoot network security issues.

Using wireshark we can view all the traffic on the interface including the packets that did not reach the destination [8]. Wireshark is similar to tcp dump the enhancement in wireshark is that this has a user-friendly GUI and it has inbuilt packet sorting and filtering functionalities.

Some of the features of wireshark are it can analyze data from the wire over the network connection or from already captured data packets, it supports live data reading and analysis, it uses display filters to organize the data displayed, new or additional protocols can be scrutinized by creating new plugins, it can also capture raw USB traffic.

V. RESULTS AND OBSERVATIONS

The below figures are the observed evaluation and results of the vulnerabilities during the various VAPT tests. These tools were run on various web applications. Some of the web applications are Apache Tomcat webserver, OWASP vulnerable servers and websites and many embedded system's web applications.

Shown below in Fig.2. is the results from scanning the web application for open ports through Nmap. Using this and http method scripts we can accurately find the ports and services they are used for and also to find if they are open/closed/filtered. This is the most crucial part as this determines the ports through which a system can be attacked, and the test performed further aim to ensure these ports are extremely protected.

Fig.2. Results from port scanning.

A storm test was performed on the web applications and was found that there where vulnerabilities and the system was not able to withstand high range of network packets as in Fig.3 and Fig.4. A solution to this would be to ignore of just drop such packets when it detects a denial of service attack. Achilles is an efficient storm testing tool as it bombards the system with the packets which helps us identify how efficient our system is in traffic handling.

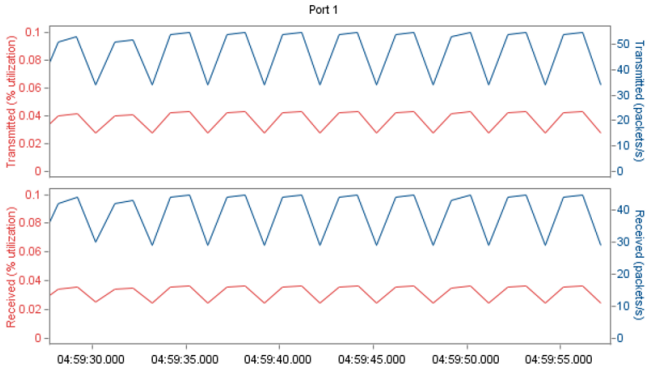
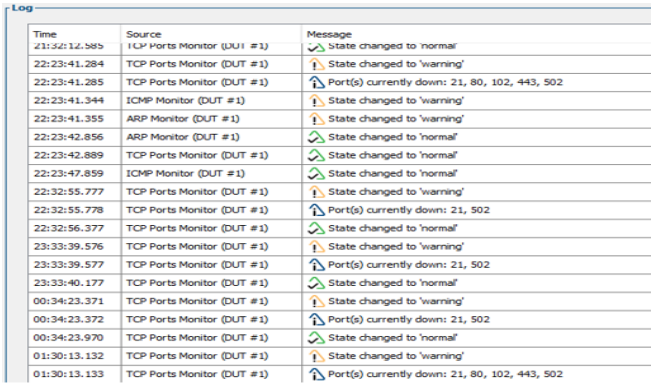


Fig.3. Instability observed through Achilles storm test



Time	Source	Message
22:23:41.284	TCP Ports Monitor (DUT #1)	State changed to 'warning'
22:23:41.285	TCP Ports Monitor (DUT #1)	Port(s) currently down: 21, 80, 102, 443, 502
22:23:41.344	ICMP Monitor (DUT #1)	State changed to 'warning'
22:23:41.355	ARP Monitor (DUT #1)	State changed to 'warning'
22:23:42.856	ARP Monitor (DUT #1)	State changed to 'normal'
22:23:42.889	TCP Ports Monitor (DUT #1)	State changed to 'normal'
22:23:47.859	ICMP Monitor (DUT #1)	State changed to 'normal'
22:32:55.777	TCP Ports Monitor (DUT #1)	State changed to 'warning'
22:32:55.778	TCP Ports Monitor (DUT #1)	Port(s) currently down: 21, 502
22:32:56.377	TCP Ports Monitor (DUT #1)	State changed to 'normal'
23:33:39.576	TCP Ports Monitor (DUT #1)	State changed to 'warning'
23:33:39.577	TCP Ports Monitor (DUT #1)	Port(s) currently down: 21, 502
23:33:40.177	TCP Ports Monitor (DUT #1)	State changed to 'normal'
00:34:23.371	TCP Ports Monitor (DUT #1)	State changed to 'warning'
00:34:23.372	TCP Ports Monitor (DUT #1)	Port(s) currently down: 21, 502
00:34:23.970	TCP Ports Monitor (DUT #1)	State changed to 'normal'
01:30:13.132	TCP Ports Monitor (DUT #1)	State changed to 'warning'
01:30:13.133	TCP Ports Monitor (DUT #1)	Port(s) currently down: 21, 80, 102, 443, 502

Fig.4. vulnerabilities observed for a storm test

When the web applications were subjected to a vulnerability scan as in Fig.5 several issues were found against the OWASP top 10 which will need immediate attention. Fig. 6 shows a vulnerability identified through Nessus.

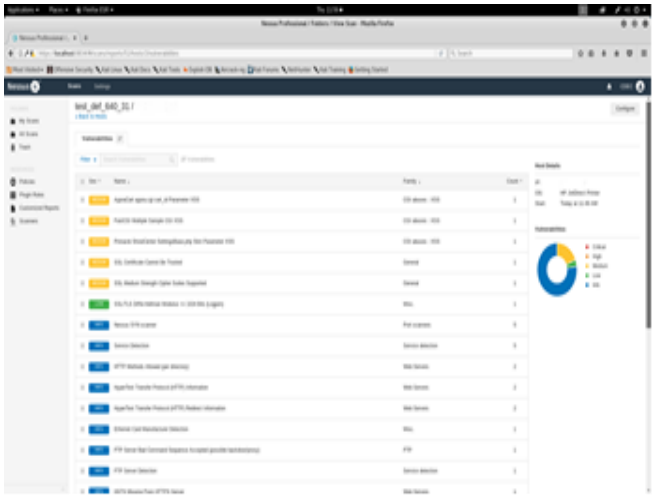


Fig.5. Security flaws observed during Nessus Scan.

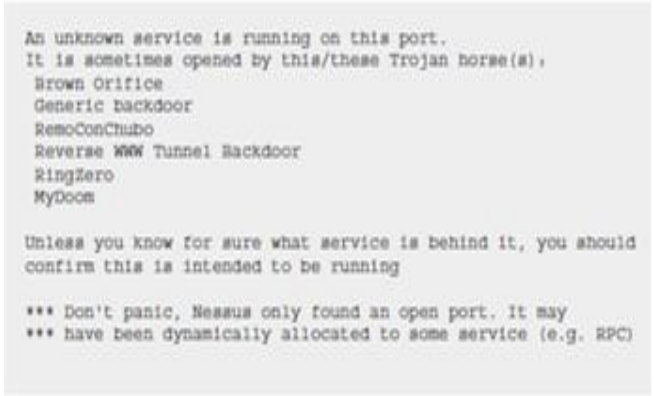


Fig.6. Vulnerability found a Nessus vulnerability test.

The Burp suite tool was able to identify several issues on performing web applications security tests. This tool performs a number of attacks like man in the middle, fuzzing, and it extensively tests for the OWASP vulnerabilities and sets the severity of these issues in correspondence to it. The Fig.7. shows a snippet of one such test results.

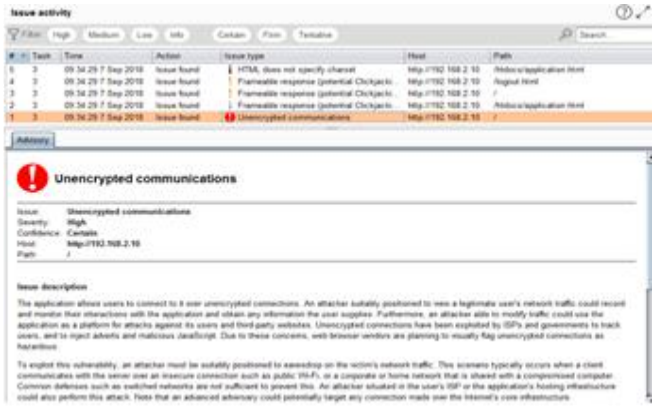


Fig.7. Test results observed from Burp Suite

The wireshark network capture show in in Fig.8. helps the tester understand if the system is under attack from a attacker or not. We can identify the malicious network traffic from a normal network traffic. This helps us to monitor and take immediate action on all the above-mentioned attacks. Network monitoring should be a crucial part of any test attack performed.

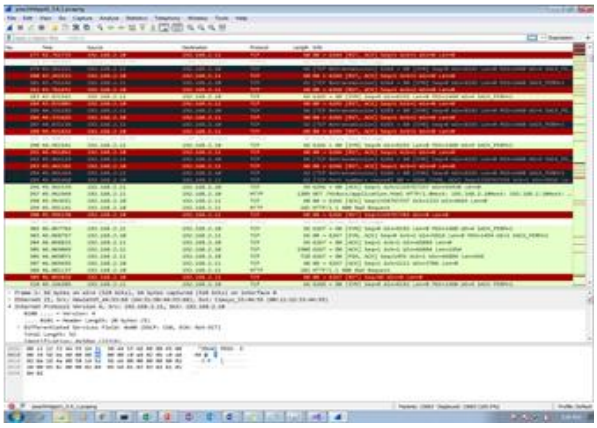


Fig.8. Network capture through wireshark.

IV. CONCLUSION

Though the performed vulnerability assessment and penetration tests were able to find most of the known vulnerabilities, these tests are alone not enough to certify a system as risk-free. To explore all the aspects of more specialized and system-specific tests must be performed which can include manual code analysis and writing custom scripts. We need to include Fuzz testing as a part of these vulnerability assessments and penetration tests to find unknown vulnerabilities present in a system. It is advised to include fuzz testing to the standard testing procedure to obtain an intrusive testing and to make sure all the loopholes are found. My future focus is to include fuzzing along with the already existing test suite of tools or testing techniques to improve the efficiency of the assessments and the results we obtain. To do so it is necessary to a thorough understanding of the system under test. When we use an open sourced tool it is of utmost importance to ensure that the fuzzer can parse deep into the system and try all the exploits. We will be using an open source fuzzing tool called Peach Fuzzer to continue with the research.

REFERENCES

1. AL-Ghamdi and Abdullah Saad AL-Malaise, “A Survey on Software Security Testing Techniques,” proceedings of ijcst conference, 2016.
2. B. Arkin, S. Stender and G. McGraw, 2005, ‘Software penetration testing’, *IEEE Security & Privacy*, vol. 3, no. 1, pp. 84-87.
3. Shah. Sugandh. & B.M. Mehtre, “A Modern Approch to CyberSecurity Analysis Using Vulnerability Assessment and Penetration Testing,” Proceedings of 2013 NCRTCST, Hyderabad (A.P), India, 2014.
4. Buja, G., Bin AbdJalil, K., BtHjMohd Ali, F.; Rahman, T.F.A., “Detection model for SQL injection attack: An approach for preventing a web application from the SQL injection attack,” proceedings of 2014 IEEE Symposium on Computer Applications and Industrial Electronics (ISCAIE).
5. OWASP Top 10 Application Security Risks, 2017, Information available from: https://www.owasp.org/index.php/Top_10-2017_Top_10 accessed at 11:30 hrs on 10-01-2019.
6. PortSwigger. (n.d.). Burp Suite, 2018, Information available from: <https://portswigger.net/burp/> accessed at 09:50 hrs on 17-01-2019.
7. Wireshark network protocol analyzer, 2018, Information available from:<https://www.wireshark.org/> accessed at 14:40 hrs on 03-01-2019.
8. Sandhya S, SohiniPurkayastha, Emil Joshua, Akash Deep, “Assessment of Website Security by Penetration Testing Using Wireshark,” proceedings of 2017 International Conference on Advanced Computing and Communication Systems (*ICACCS* -2017).

AUTHORS PROFILE

Gitanjali Simran, T Completed her PG at Bannari Amman Institute of Technology Sathyamangalam, India

Dr. D. Sasikala, has 19 years of experience and currently working as professor in the department of Computer Science and Business Systems.