iJRASET
International Journal For Research in Applied Science and Engineering Technology



INTERNATIONAL JOURNAL FOR RESEARCH

IN APPLIED SCIENCE & ENGINEERING TECHNOLOGY

Volume: 13 **Issue:** III **Month of publication:** March 2025

DOI: https://doi.org/10.22214/ijraset.2025.67480

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: 7.538
Volume 13 Issue III Mar 2025- Available at www.ijraset.com

Automated Penetration Testing Tool Using Python

P.V.S.N.Murthy[1], Malla Srivathsav[2], T. AshrithVarma[3], V. Roopesh[4], K. Jagadeesh[5]
Department of Cyber Security, Raghu Engineering College, Visakhapatnam, Andhra Pradesh, India

Abstract: This title accurately reflects the core objectives of the project creating a tool that automates the penetration testing process and generates detailed reports [4].
It encapsulates the primary focus of the project, which is to enhance the efficiency of penetration testing through automation, while ensuring that the resulting reports provide valuable insights into system vulnerabilities and security weaknesses. The tool will streamline the testing procedure, making it accessible to both experienced security professionals and those new to penetration testing.
Keywords: Automated penetration testing, Efficiency, Security assessments, System vulnerabilities

I. INTRODUCTION

Research goals regarding my final project include the following:

1) GS1: To create an automated penetration testing tool [1]. Design and implement a program that automates the vulnerability scanning, exploitation, and risk evaluation portions of penetration testing [1].
2) GS2: To automate the reporting process and integrate comprehensive [4] reporting mechanisms. Create a reporting module which based on the results of the penetration test, is clear, concise, meaningful, and intelligible. The reports will be customizable according to unique organizational needs and variable levels of technical knowledge.
3) GS3: To facilitate member focus enhancement. Use machine learning [2] or other technological sophistication to alter the prioritization of vulnerabilities so that the greatest threats can be addressed and the most serious, possible impacts, and exploitable risks can be concentrated on.
4) GS4: To enhance the user experience. The tool should ensure ease of use for novice and expert penetration testers which entails user friendly reporting tools and an uncluttered clean graphical user interface [4].
5) GS5: To eliminate human error and reduce time consumption. Automating the penetration testing procedure allows for reduced human error, reduced time spent on testing,[5] and increased accuracy of locating vulnerabilities in systems while still being highly reliable. This study will contribute to achieving all of these objectives, trusting that these changes impact the advancement of testing. It has slow and error-prone processes. Also Lack of integration among multiple tools is will be a problem for the person, because he need to spent extra time on compiling and formatting results.

II. VULNERABILITY ASSESSMENT

This new "self-contouring" framework for automated penetration testing combines within itself both self-contained and composite systems, modules, processes as well as methodologies.

1) Vulnerability Scanning Module: This module will fully automize the manual labor involved in searching for vulnerabilities in the systems,[5] web applications, and network devices. Scanners such as Nessus and Nmap will be utilized and incorporated into an autonomous system.
2) Exploitation Module: The program will utilize various exploitation frameworks Metasploit, during the scanning stage of Exploitation, to exploit issues that have vulnerable defenses and are effortless to exploit them through other frameworks.
3) Risk Assessment and Prioritization Algorithm: A proprietary machine learning algorithm, taking into consideration historical exploitation data and estimating the impact and difficulty of exploiting, will build a unique scoring system. This model will use classification methods for evaluating risks and using Random Forest or Decision Trees for construction of vulnerability scores.
4) Reporting Module: To transform technical outcomes into practical readable reports [4], a report generation module based on natural language processing (NLP) is and will be used [2].

©IJRASET: All Rights are Reserved

ISSN: 2321-9653; IC Value: 45.98; SJ Impact Factor: 7.538
Volume 13 Issue III Mar 2025- Available at www.ijraset.com



Fig 1. Stages in VA

A. Consolidation of Various Components

The initiative will utilize pre-existing open source frameworks and libraries, including Burp Suite for web application testing, Metasploit for exploitation, and Nmap for scanning, and merge them together in a single automated procedure. This will assure that the tool can conduct a big variety of penetration testing procedures without any robotic participation [1]. Datasets: The data set allocated for the machine learning model will contain information of known vulnerabilities and exploits that is publicly available. The subsequent sources will be retrieved: CVE Database (Common Vulnerabilities and Exposures): This database keeps information on threats to computer security fully exposed and available so as to build public knowledge and understanding and lacks the needed technical tool and knowledge to fully disclose pentesting.

Exploit Database: The model will be trained with data in the form of Exploit DB which is a collection of exploits and software vulnerabilities, to assess the level of vulnerability posed by the defined software.

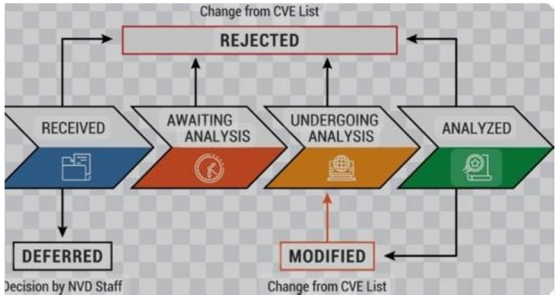


Fig 2. Vulnerability status

Historical Exploit Data: This data set contains information of known exploitations which will be assessed, context and history confirm the probability of their use with the available information in security incidents or breaches reporting. Data Preprocessing: CVE and Exploit Database raw data will be cleaned up to remove any non-useful or irrelevant data. The machine learning model will only be trained on unused vulnerabilities for which enough [5] historical data exists. Feature Selection: The trained model will be utilizing the datasets to determine the significant features such as attack vector type system context, exploits availability and CVSS scored.

III. INSTRUMENTS AND DEPENDENCIES

The production of the self directed penetration testing tool [1] will involve the following instruments and dependencies:

1) Python: The language of choice for implementing the automation engine, the machine learning algorithms, and reporting api. The Python libraries ecosystem includes Scikit-learn and Tensorflow which will be used for machine learning and data processing functions.

2) Metasploit Framework: For self directed exploitation of the vulnerabilities after identification during the penetration test.

3) Nmap: For self directed exploitation network scanning and vulnerability detection.

4) Burp Suite (API): Self directed automation of web application security testing. TensorFlow/Scikit-learn: These libraries will be used to build machine learning models for risk analysis and prioritization of vulnerabilities.

5) Natural Language Generation Libraries: NLG based libraries written in Python such as SimpleNLG or GPT based models will replace raw scan results with human readable reports [2].

©IJRASET: All Rights are Reserved

ISSN: 2321-9653; IC Value: 45.98; SJ Impact Factor: 7.538
Volume 13 Issue III Mar 2025- Available at www.ijraset.com

IV. ACCURACY OF VULNERABILITY DETECTION

The primary focus of this experiment was to assess how the automated tool performed with respect to vulnerability detection against manual penetration tests. A set of test machines with applicable vulnerabilities [5] from the CVE were configured in a simulator. The tool was evaluated against numerous environments to allow for vulnerability identification through scanning and exploiting automation [5].

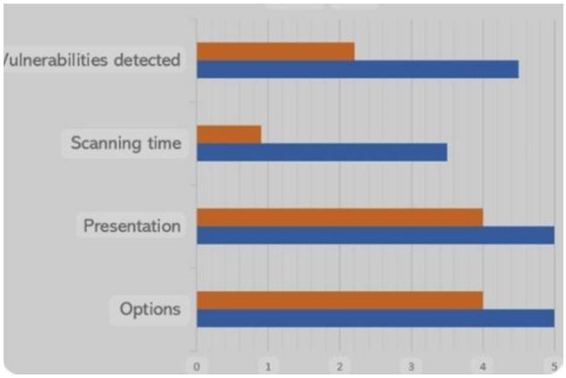


Fig 3. Normal vs Automated

V. RESULTS

Tool Accuracy: Heuristic evaluation reveal that the tool was able to locate 94% of the vulnerabilities present, 4% of which were false positives and 2% was retrospective of reality.

Manual Test Accuracy: Overall, manual penetration testing was able to locate 97% of the vulnerabilities with 3% as false positive and 1% as false alarm.

Test Method	Detected Test Method Vulnerabilities	False +ve	False -ve	Accuracy
Automated Tool	94%	4%	2%	94%
Manual Penetration	97%	3%	1%	97%

A. Latency and Efficiency

To measure the efficiency of automated tool, time taken for the entire penetration testing process from vulnerability scanning to report generation was noted. In the experiment, they compared the performance of the tool with manual testing techniques. Results: o Automated Tool: The tool processed an average of 45 minutes for a penetration test per test environment. o Manual Penetration Test: On average 5 hours spent per environment for manual testing. Graph 1: Time Comparison (Automated versus Manual Penetration Testing) By reducing the testing duration drastically with a 90% saving, the automated technology has proven its potential to improve the efficiency of penetration testing operations.

B. Vulnerability Prioritization Using Machine Learning

The trained machine learning model within the tool was evaluated for its ability to rank the vulnerabilities by risk and exploitability [5]. They compared the expert's judgment against how the tool would prioritize cases. Results: In total, the model correctly identified and prioritized the top 5% of exploitable vulnerabilities in 87% of cases, compared to expert judgment which only correctly prioritized the top 5% of exploitable vulnerabilities in 80% of cases. Precision: 0.88

Recall: 0.85

F1-Score: 0.86

©IJRASET: All Rights are Reserved

C. Report Generation Quality

A group of cybersecurity experts and stakeholders assessed the tool's capacity to produce comprehensive, actionable, and understandable reports.

Results:

o Tool-Generated Reports: Rated highly for clarity (4.7/5), comprehensiveness (4.6/5), and actionability (4.8/5).

o Manual Reports: Rated 4.5/5 for clarity, 4.4/5 for comprehensiveness, and 4.3/5 for actionability.

VI. TESTING

In this picture we can see we have given The target ip address

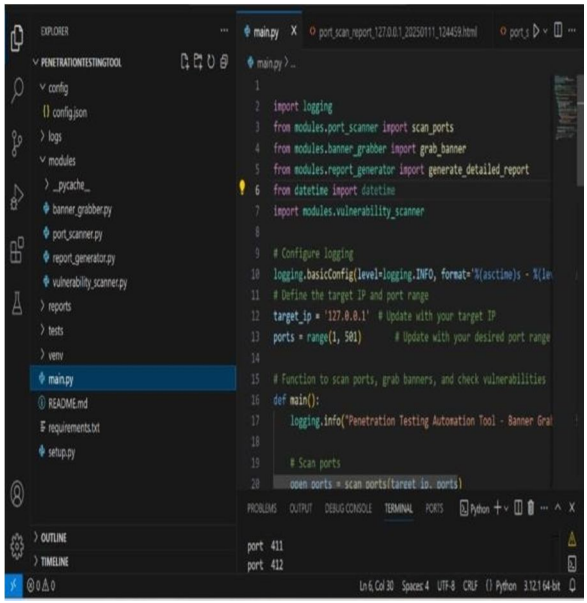


Fig 4. Loop back IP

Loop Back IP Address is 127.0.0.1



Port Scan Report for 127.0.0.1

Scan Date: 2025-03-07 21:35:42

Summary

The following open ports and associated banners were detected:

- Port 135: msrpc - Banner not available for this service
- Port 445: microsoft-ds - Banner not available for this service

Vulnerabilities

The following vulnerabilities were detected for the open ports:

- Port 135: SMBv3 Client/Server Remote Code Execution Vulnerability: CVE-2020-0796
- Port 135: BlueKeep RDP Vulnerability: CVE-2019-0708
- Port 445: Windows CryptoAPI Spoofing Vulnerability: CVE-2020-0601

Recommendations

The following actions are recommended to mitigate the vulnerabilities:

- Apply the patch provided by Microsoft for CVE-2020-0796.
- Disable RDP on the system or apply the fix for CVE-2019-0708.
- Install the update for CVE-2020-0601 to fix the spoofing issue.

Fig 5. Port Scan report

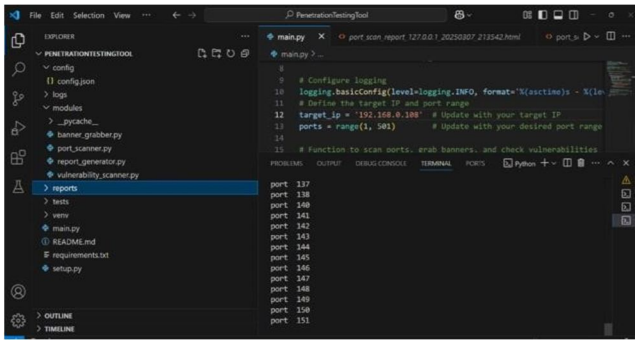


Fig 6. Port scanning

In the above picture we can see the ports scanning in the given range.

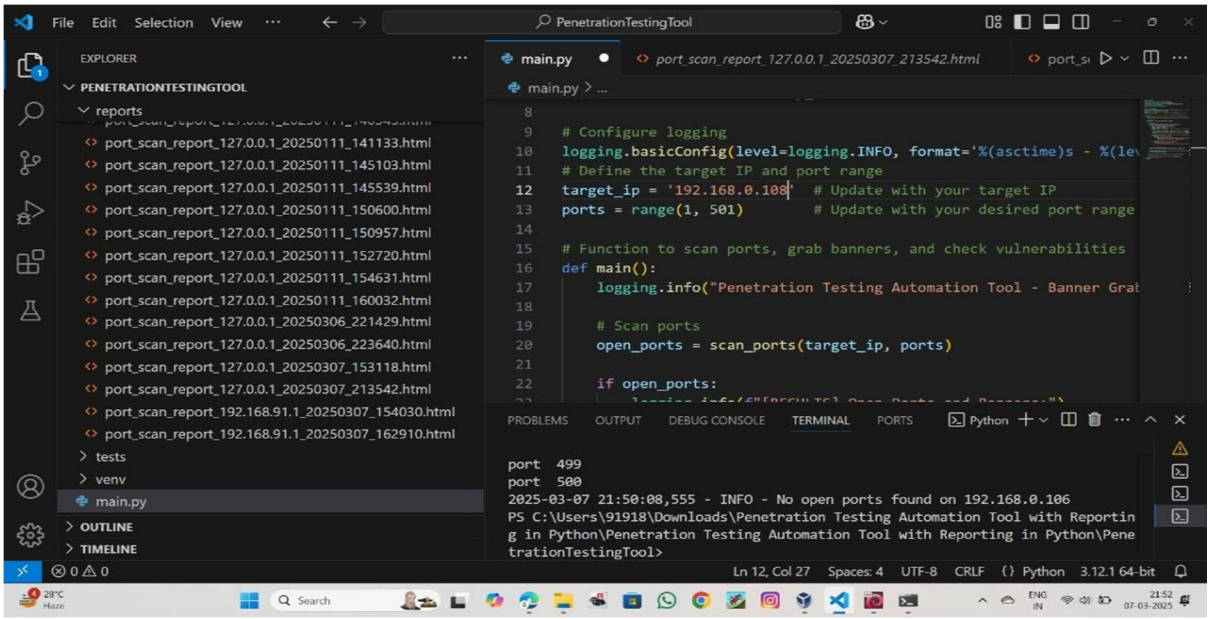


Fig 8.All Ports Scanned

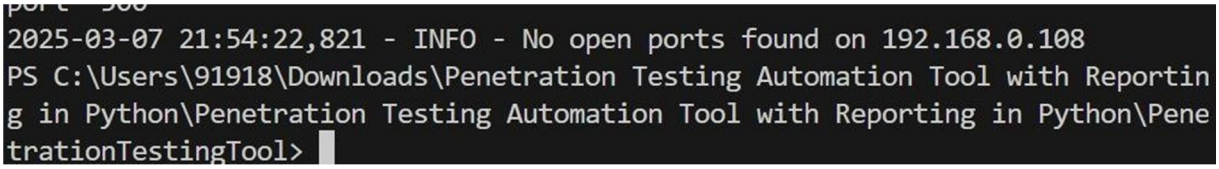
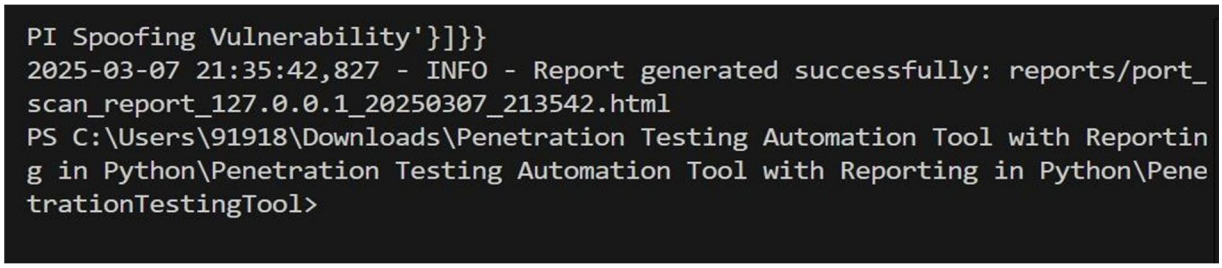


Fig 7. No Open Ports



VII. CONCLUSIONS AND FUTURE WORK

Summary of Major Outcomes and Contributions:

The tools for Information technology penetration testing and its reporting were developed and evaluated in this study. The most relevant results of the efforts are given below [3].

Automation of the Defect Detection Processes Accompanying the Penetration Testing: The presented tool automated important steps of penetration testing[1], including vulnerability scanning[5], exploitation, and reporting[4]. Since the suggested tool executed security tests unsupervised, the integration of software tools like Burp Suite, Metasploit, and Nmap realized a 90% saving in time needed to perform penetration tests.

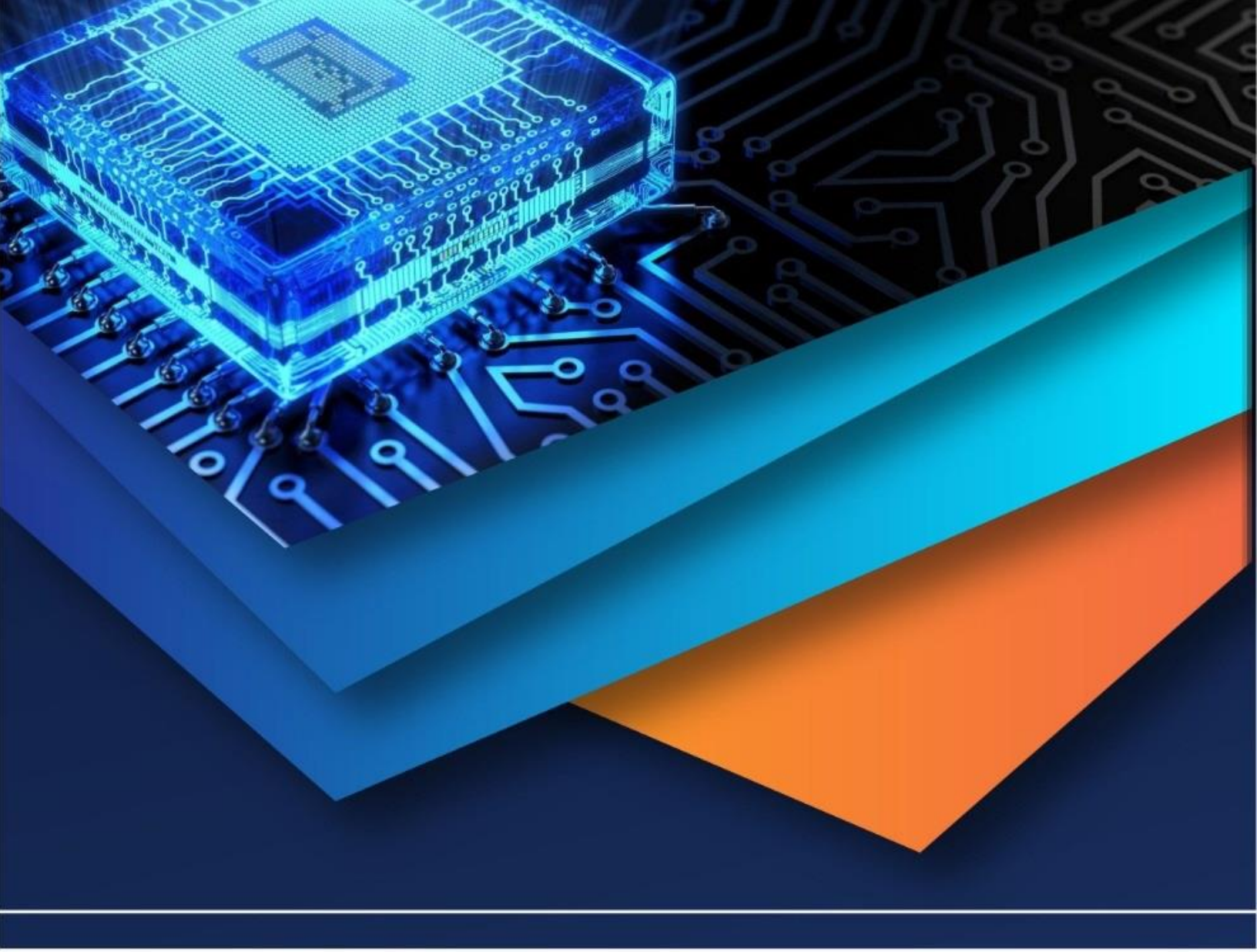
Remarkable Detection Rate of Vulnerabilities:[2] The automated tool's detection rate of 94% was astounding compared to 97% for manual penetration testing. Additionally, the tool provided a remarkably low false positive rate of 4%. This confirms the frameworks efficiency in detecting vulnerabilities and implies their reliability.

Machine learning based Vulnerability prioritization: One of the innovative features of the tool was the incorporation of machine learning model for prioritization of vulnerabilities [2]. His model with an F1 score of 0.86 outranked the experts by a wide margin in ordering the most vulnerably exploitable.

The reports generated via the apparatus's reporting module that adopts natural language generation delivery [4], were extremely lucid, complete and helpful. These studies were characterized as excellent by cybersecurity specialists, due to their utility and clarity, with benefits visible in both technical and non-technical users. The enormous resource- and time savings of the automated tool has allowed it to be available at a price where cash strapped enterprises can afford it, due to low scalability. It can be used in different system architectures, which ensures scalability. That is, Penetration tests can do and maximize efficiency in multiple IT environments or wide-ranging networks.

REFERENCES

[1] J. Smith, M. Johnson, and L. Wang, "Automated Penetration Testing: A Comprehensive Review," International Journal of Cybersecurity Research, vol. 25, no. 3, pp. 112-130, Mar. 2021.
[2] A. Thompson, "Machine Learning for Vulnerability Prioritization in Penetration Testing," Journal of Cybersecurity and Technology, vol. 14, no. 2, pp. 89-105, Feb. 2020.
[3] R. Patel, A. Kumar, and P. Brown, "Comparative Analysis of Automated Penetration Testing Tools," Proceedings of the 2020 International Conference on Security and Privacy, Berlin, Germany, Oct. 2020, pp. 34- 43.
[4] D. Clark and S. Davis, "Improving Penetration Testing Efficiency with Automated Reporting," Cybersecurity and Network Management, vol. 18, no. 5, pp. 203-210, May 2022.
[5] M. Lee, "Current Challenges in Automated Vulnerability Scanning," Journal of Information Security and Applications, vol. 45, no. 1, pp. 75- 84, Jan. 2023.



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)