International Research Journal on Advanced Engineering and Management
https://goldncloudpublications.com
https://doi.org/10.47392/IRJAEM.2025.044

e ISSN: 2584-2854
Volume: 03
Issue:02 February 2025
Page No: 274-282

A Comparative Study of Penetration Testing Methodologies and Tool Utilization in Cybersecurity

Uraj U. Sahu[1], Jitendra B Upadhyay[2], Satish Prakashrao Pise[3]

[1]PG-Student, Shrimad Rajchandra Institute of Management and Computer Application, Uka Tarsadia University, Bardoli, Gujarat, India.

[2]Assistant Professor, Shrimad Rajchandra Institute of Management and Computer Application, Uka Tarsadia University, Bardoli, Gujarat, India.

[3]Assistant Professor, DKTE institute of textile and engineering, Rajwada, Ichalkaranji, Maharashtra, India.

Email ID: *sahuu5249@gmail.com[1], jbupadhyay@utu.ac.in[2], satish.pise@dkte.ac.in[3]*

Abstract

The growth in volumes of data traded on the internet requires network protection. The customers demand their data to be held in secrecy, intact, and available only to those who have access. Hence, three security requirements for maintaining such networks would include confidentiality, integrity, and authenticity (CIA). Traditionally, organizations were protecting their physical assets. However, in today's digital world, they have to protect themselves from internal as well as external cyber threats. This has, therefore, become a necessity to ensure proactive methods in assessing and strengthening security systems. One of the methods that are simulating real-time cyberattacks is penetration testing. These test vulnerabilities that would otherwise be left unnoticed by other methods. Organizations can learn about such risks, address them, and therefore become more prepared to respond to new threats as their defenses strengthen and their cybersecurity becomes stronger.

Keywords: *Confidentiality Integrity Availability (CIA); Cybersecurity; Ethical Hacking; Network Security; Penetration Testing*

1. Introduction

Kumar Shravan[1], among others, continues elaborating on what Pen-Testing is. “Pen-testing means verifying whether the security in place by an organization meets the CIA standard, wherein data used by organizations remains confidential and not tempered with, but available when needed”. There are diverse manners of pen-testing, like. Therefore, penetration testing is considered an important methodology for the identification of vulnerabilities and remediation through improving the strength of systems to face various cyber-attacks. The whole process is checking the systems from both internal and external perspectives with the help of different types of testing approaches such as Black Box Testing, White Box Testing, and Gray Box Testing. Penetration Testing, also known as Pen-Testing or Ethical Hacking, is a security assessment technique simulating an attack on a system, network, or application to identify possible vulnerabilities before malicious actors can exploit them. According to Irfan Yaqoob et all[2]., this form of penetration testing is “unauthorized access to the systems without the use of valid credentials for identifying weaknesses that might be exploited by the attackers”. Processes help organizations assess how effective their current security measures are and take necessary corrective actions to mitigate risks. It understands the threat source as well as makes an impact assessment of an attack and recommends how to strengthen security. Penetration testing is always a dynamic process because its nature is directed by changes in cyber threats that are emerging currently. For instance, the “Target data breach” that occurred in 2013(NBC NEWS) is a reason why penetration testing is very important-it shows the need for periodic penetration testing. The attackers used several weak spots in the network of the company to steal millions of customer records. Had there been proactive penetration testing, all those vulnerabilities in terms of inadequate segmentation and weak third-party access controls would have been discovered and mitigated. This example illustrates the need for penetration testing in safeguarding organizational assets and maintaining a robust security infrastructure against evolving threats.

2. Literature Review

The paper by Kumar Shravan, Bansal Neha, and Bhadana Pawan [1], introduces penetration testing as a method to validate systems against CIA standards (Confidentiality, Integrity, Authentication) by identifying vulnerabilities. It elaborates on the goals of penetration testing, including enhancing system security, identifying weaknesses, and improving organizational infrastructure. The process includes planning, information gathering, exploitation, and reporting. Testing types fall into three main categories: Black-box, White-box, and Grey-box testing, depending on the knowledge level of the tester. Penetration testing phases are further divided into pre-attack (reconnaissance), attack (exploitation), and post-attack (system restoration). Network attacks and social engineering are two testing methods the paper discusses; however, limitations include time constraint and missed vulnerabilities. Minimum requirements for penetration testing, such as approvals, scope definition, and service-level agreements, are highlighted. Irfan Yaqoob et al[2], discuss the need to identify and mitigate system vulnerabilities, such as outdated software or weak passwords. Vulnerability Assessment (VA) is defined as identifying and remedying weaknesses, while penetration testing simulates real cyberattacks to analyze system defenses. The paper focuses on Black-box, White-box, and Grey-box testing types, emphasizing the need for well-defined scope and agreements for effective testing. While recognizing time and resource constraints, the authors stress the importance of updating tools and methods to address evolving threats. Parvin Ami and Ashikali Hasan[3] propose a systematic seven-phase process: preparation, anonymity, footprinting, vulnerability analysis, exploitation, reporting, and advisory. The model covers an audit process, from data collection to offering remediation advice. Types of testing, including Black-box and White-box, are discussed; these are aimed at simulating real-world attacks and looking inside the system structure. The paper does mention issues such as budget and incomplete outcomes but still calls for risk assessment, detailed documentation, and practical recommendations. In the research of Mamilla Sushmitha Reddy[4], there is the great need to understand penetration testing with the present rate of increasing data breaches. This paper traced how penetration testing developed from "tiger teams" of the 1960s to present methodologies, in that it differentiated it from a vulnerability assessment with comprehensive simulation through both manual and automated techniques on real-world attacks. The paper will discuss the objectives of penetration testing, types, which include network, application, client-side, and social engineering, and models such as Flaw Hypothesis and Attack Tree for adversarial simulations. The role of penetration testing in the strengthening of cybersecurity through detailed analysis of processes and tools is emphasized. After this, the comparative study will begin, where I will focus on comparing the tools, phases, and types of different penetration testing models.

3. Comparative Study

3.1. Phases of Penetration Testing

A comparison table of various authors' phases of penetration testing is provided, which shows an overview of the phases proposed by different authors regarding penetration testing. This comparison table also depicts what the authors contribute in particular or provide extra apart from the above-defined phases. Further comparing these phases emphasizes individual approaches and focuses of every study, for instance, simulating stealth attacks, real-time guidance, and documentation of vulnerabilities. Comparing these phases helps in bringing out the diversity in methodologies within the field of penetration testing. From Table 1, we see that the three main sets of phases-the one by Parvin Ami and Ashikali Hasan, Irfan Yaqoob et al., and Kumar Shravan, Bansal Neha, and Bhadana Pawan-differ at the level of penetration testing. All authors separate penetration testing into distinct phases; however, such phases are grounded on the authors' differing priorities and methodologies.

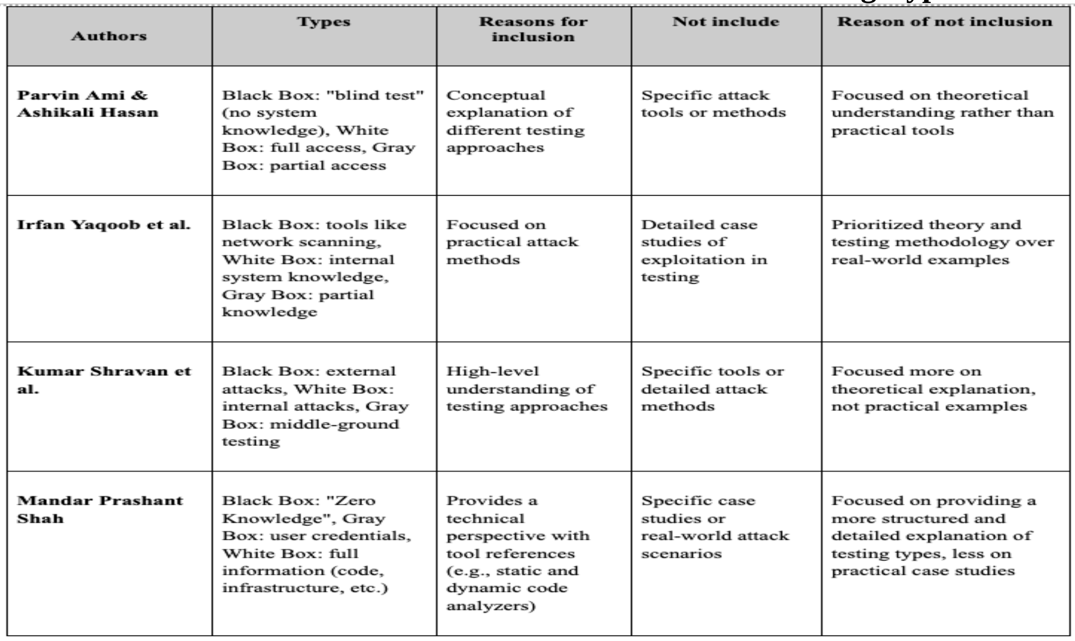
Table 1 Various Authors' Phases of Penetration Testing

Authors	**phases**	**Additional contribution**
Parvin Ami Ashikali Hasan	Planning,, Footprinting, Analysis, Advisory, Reporting, Exploitation	**Animosity**: Focuses on simulating real-world stealth attacks. **Advisory**: Emphasizes real-time guidance, unlike others
Irfan Yaqoob Syed Adil Hussain Saqib Mamoon Nouman Naseer Jazeb Akram	Planning, Reconnaissance, Exploration, Vulnerability Assessment,	**Exploration**: Focuses on detailed investigation of targets. **Vulnerability Assessment**: Emphasizes documenting vulnerabilities explicitly.
Kumar Shravan , Bansal Neha , Bhadana Pawan	Planning, Discovery, Exploit, Reporting	Combines data gathering (Footprinting/Reconnaissance) into "Discovery" for simplicity. Streamlines the process to essential phases.

Kumar Shravan, Bansal Neha, and Bhadana Pawan[1] break the process into four stages: Planning, Discovery, Exploitation, and Reporting. It has reduced the process to some extent by amalgamating the two activities—Footprinting and Reconnaissance—under a single activity, called Discovery. The only negative impact is that it has lost the initial depth in the information gathering activity. Table 1 indicates quite an amount of effort has been put into core testing activities, thus suitable for rapid evaluation. Phase set of Irfan Yaqoob et al[2] can be classified under five: Planning, Reconnaissance, Exploration, Vulnerability Assessment, and Exploitation. This model includes the Exploration phase, which is a close-in study of targets. This way, penetration testers can probe in-depth the weaknesses of systems under test. This set of phases explicitly focuses on Vulnerability Assessment, ensuring that discovered weaknesses are documented for remediation. Added focus on Exploration and Vulnerability Assessment is clearly represented in Table 1, where this approach is further emphasized. Parvin Ami and Ashikali Hasan[3] set of phases is totally divided into seven phases: Planning, Footprinting, Analysis, Advisory, Reporting, Exploitation, and Animosity. In fact, a really fascinating set of phases because an Animosity phase, simulating how stealth attacks occurring in real-world environments, might show how it is possible that an attacker goes about without having ever been spotted. Their Advisory phase brings it immediate advice directly to the teams of security - proactive. This is in contradistinction from other phases which are somehow static and documentation-oriented. From Table 1, the existence of Animosity and Advisory phases reflects that the authors proposed a holistic approach of penetration testing proactive. If comparing these phases from Table 1, then it is observable that each set has its strength besides potential gaps. Phases by Parvin Ami and Ashikali Hasan are robust for simulating real attack strategies besides offering detailed advisories; however, seven phases might be difficult to handle by small teams or resource-scarce organizations. Irfan Yaqoob et al. have a very well-balanced set of phases with risk assessment and vulnerability documentation but perhaps may not be so feasible in high-pace environments where the priority is on speed rather than details. Kumar Shravan, Bansal Neha, and Bhadana Pawan have streamlined phases but perhaps missing granularity as seen in the other sets, perhaps leaving some vulnerabilities and inadequate post-test analysis. As shown in Table 1, comparison of these phases can be made with the understanding that each set of phases is useful for penetration testing. It largely depends on the unique needs of the organization, the resources, and the depth to which it needs to test.

3.2. Types of Penetration Testing

Table 2 Different Authors Toward Penetration Testing Types

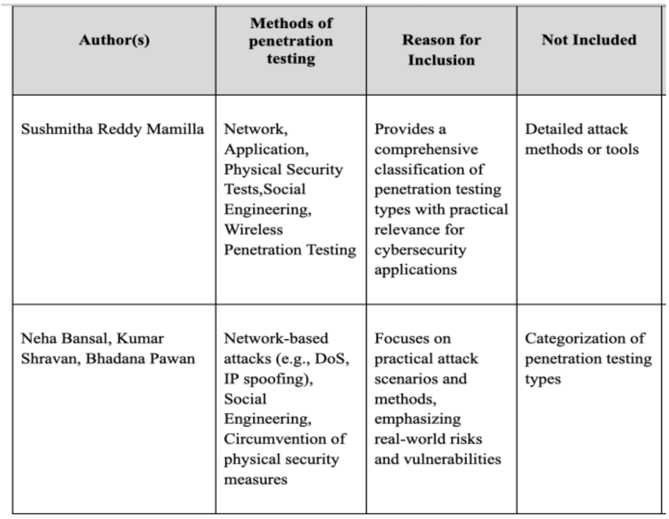


Authors	Types	Reasons for inclusion	Not include	Reason of not inclusion
Parvin Ami & Ashikali Hasan	Black Box: "blind test" (no system knowledge), White Box: full access, Gray Box: partial access	Conceptual explanation of different testing approaches	Specific attack tools or methods	Focused on theoretical understanding rather than practical tools
Irfan Yaqoob et al.	Black Box: tools like network scanning, White Box: internal system knowledge, Gray Box: partial knowledge	Focused on practical attack methods	Detailed case studies of exploitation in testing	Prioritized theory and testing methodology over real-world examples
Kumar Shravan et al.	Black Box: external attacks, White Box: internal attacks, Gray Box: middle-ground testing	High-level understanding of testing approaches	Specific tools or detailed attack methods	Focused more on theoretical explanation, not practical examples
Mandar Prashant Shah	Black Box: "Zero Knowledge", Gray Box: user credentials, White Box: full information (code, infrastructure, etc.)	Provides a technical perspective with tool references (e.g., static and dynamic code analyzers)	Specific case studies or real-world attack scenarios	Focused on providing a more structured and detailed explanation of testing types, less on practical case studies

Table 2 Comparison of kinds of penetration tests discussed by several authors and focusing in particular on three kinds of: Black Box; White Box and Gray Box types. Reasons provided for the above three approaches for the three different authors that either conceptually explain the respective kind or points towards the applied tool, conversely indicating omissions, say real-life cases or attack technique. This table shows the diversity in how authors address penetration testing, with some focusing on theory and others on practical application, but none fully integrates both. The comparative study based on Table 2 sheds light on the approaches of different authors toward penetration testing types and the existing gaps in their methodologies. Kumar Shravan, Bansal Neha, and Bhadana Pawan [1] provide a condensed overview of the testing approach, focusing on external, internal, and partial access types of testing. While their explanation is well-crafted, it lacks practical implementations and specific tools for practitioners. Parvin Ami and Ashikali Hasan [3] focus on the theoretical framework of Black Box, White Box, and Gray Box testing, providing a conceptual understanding based on the tester's knowledge of the system. While their approach offers a solid grasp of basic concepts, it does not include examples of tools or practical references. Irfan Yaqoob et al. [2] integrate practical aspects such as network scanning tools with Black Box testing, emphasizing methodologies. However, their omission of detailed case studies and exploitation techniques limits the work from reflecting actual exposure. Mandar Prashant Shah [4] is notably valuable for including static and dynamic code analyzers and their practical applicability for professionals. However, his work lacks real case studies or attack scenarios, which reduces its contextual relevance. All authors have gaps in integrating theoretical concepts with practical tools, limited discussion on advanced techniques, and a lack of detailed real-world case studies to contextualize the methodologies. Although each author makes a significant contribution to a particular area of penetration testing, their work does not form a comprehensive framework that connects theory and practice. This suggests the need for a holistic approach combining conceptual understanding, practical tool usage, and case studies to fulfill the diverse needs of penetration testing in both academic and professional environments.

3.3. Methods of Penetration Testing

Table 3 Different Authors Provide Their Methods for Penetration Testing



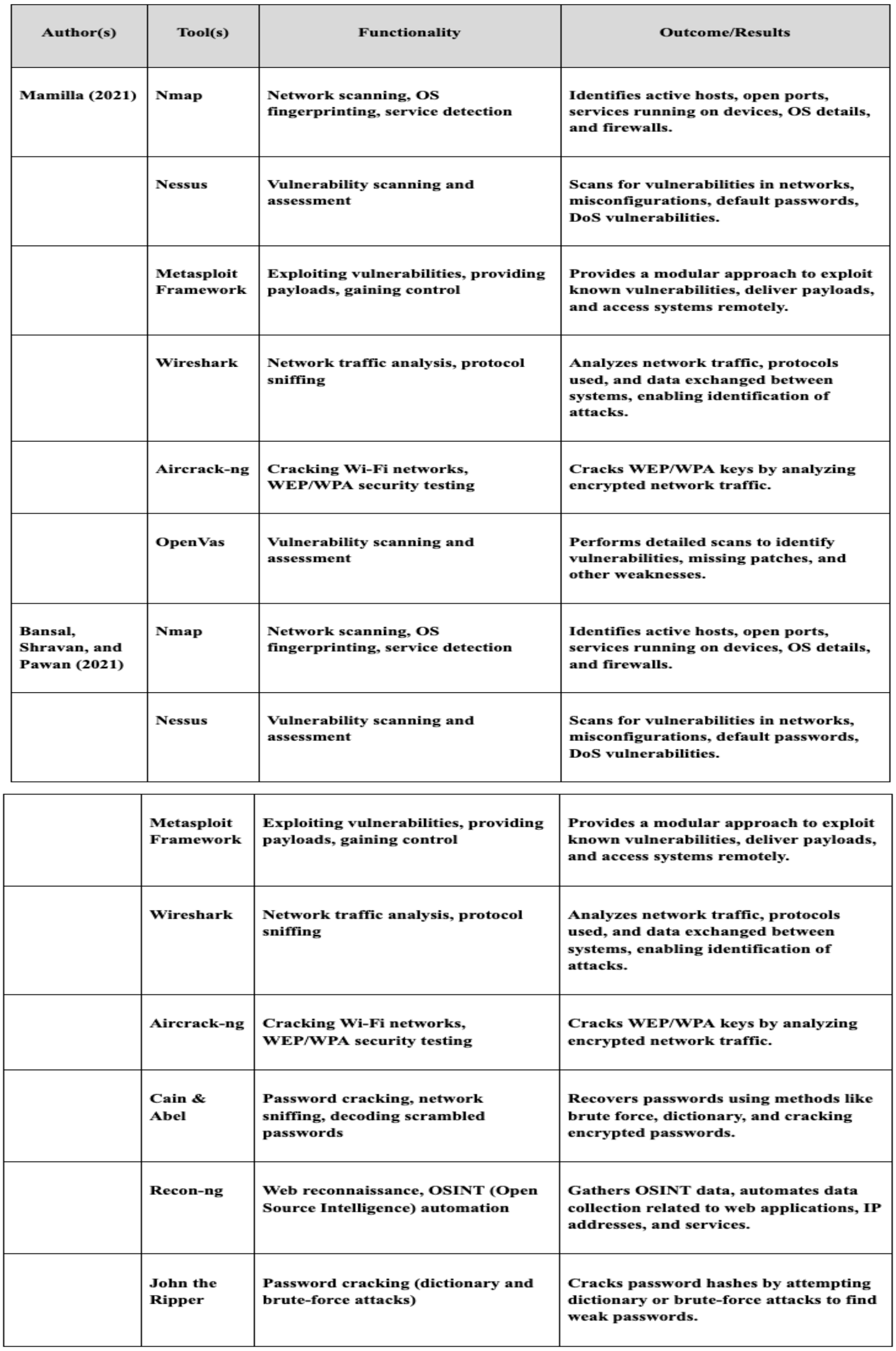
Author(s)	Methods of penetration testing	Reason for Inclusion	Not Included
Sushmitha Reddy Mamilla	Network, Application, Physical Security Tests,Social Engineering, Wireless Penetration Testing	Provides a comprehensive classification of penetration testing types with practical relevance for cybersecurity applications	Detailed attack methods or tools
Neha Bansal, Kumar Shravan, Bhadana Pawan	Network-based attacks (e.g., DoS, IP spoofing), Social Engineering, Circumvention of physical security measures	Focuses on practical attack scenarios and methods, emphasizing real-world risks and vulnerabilities	Categorization of penetration testing types

Table 3 compares two studies on penetration testing. Sushmitha Reddy Mamilla classified penetration test type but lacked specifics on specific attacking methods and has been excluded in practical studies. Neha Bansal, Kumar Shravan, and Bhadana Pawan were emphasizing real-time attack methods that could be DoS and social engineering and have been excluded from reviews, which were solely focused on the classification. As per table 3, the study by Neha Bansal, Kumar Shravan, and Bhadana Pawan[1] highlights specific attack methods such as network-based methods viz. Denial of Service (DoS) and IP spoofing, social engineering, and physical security circumvention. This makes it a practical study and portrays real-world risks and vulnerabilities, thereby giving actionable insights on how cyber insecurity challenges could be addressed. On the other hand, As per table 3,This Sushmitha Reddy Mamilla[4] study would focus on categorizing penetration testing methods, that is, network, application, physical security, social engineering, and wireless testing. Her work is structured well, making it best suited for grasping the vast categories of penetration testing and how they are implemented. The difference would be in its focus. Work by Mamilla [4] is theoretical classification-oriented, where the study work by Bansal, Shravan, and Pawan[1] is actually practical application-type. Together, they represent a comprehensive value for both people who want fundamental knowledge about penetrating testing types but also for somebody who wants information about attack ways and vulnerabilities as well. When comparing the two studies, there is a point of difference- focus. Mamilla's study was theoretical and based on classification so that it structured the framework into understanding the kinds of penetration tests. This has made it ideal to be used educationally or just to develop the broadest perspective of the area. The practical and application study by Bansal, Shravan, and Pawan is much more specific attack methods and its implications. This makes it more appropriate for professionals interested in solving real-world cybersecurity challenges.

3.4. Tools of Penetration Testing

The table 4 compares the tools and their functionalities in penetration testing as identified in the studies by Bansal, Shravan, and Pawan [1] and Mamilla [4]. Both studies use a variety of tools to identify vulnerabilities and assess security in networks and systems.The penetration testing tools employed by Bansal, Shravan, and Pawan [1] and Mamilla [4] are compared based on similarities and differences. These studies utilize some widely recognized tools, including Nmap, Nessus, Metasploit Framework, Wireshark, and Aircrack-ng for critical functionalities such as network scanning, vulnerability assessment, traffic analysis, and Wi-Fi security testing. These tools assist in detecting active hosts, open ports, misconfigurations, encrypted network traffic vulnerabilities, and exploitation for system access delivery. The study conducted by Bansal, Shravan, and Pawan [1] mainly focuses on core tools, offering a good understanding of penetration testing and its commonly used methods. The study by Mamilla [4], however, is broader as it includes more tools like Cain & Abel, Recon-ng, and John the Ripper for cracking passwords. These tools have capabilities that include OSINT for web-based reconnaissance and better automated data acquisition from web-based applications. The addition of these tools makes the identification of weak passwords more holistic and comprehensive, as well as gathering actionable intelligence on web services, making the study more robust in addressing different cybersecurity challenges. Further, while both studies provide useful insights, they do not mention tools like Burp Suite and Hydra, which are often considered essential in the penetration testing framework that goes beyond PWK.

Table 4 Compares the Tools and Their Functionalities in Penetration Testing



Author(s)	Tool(s)	Functionality	Outcome/Results
Mamilla (2021)	Nmap	Network scanning, OS fingerprinting, service detection	Identifies active hosts, open ports, services running on devices, OS details, and firewalls.
	Nessus	Vulnerability scanning and assessment	Scans for vulnerabilities in networks, misconfigurations, default passwords, DoS vulnerabilities.
	Metasploit Framework	Exploiting vulnerabilities, providing payloads, gaining control	Provides a modular approach to exploit known vulnerabilities, deliver payloads, and access systems remotely.
	Wireshark	Network traffic analysis, protocol sniffing	Analyzes network traffic, protocols used, and data exchanged between systems, enabling identification of attacks.
	Aircrack-ng	Cracking Wi-Fi networks, WEP/WPA security testing	Cracks WEP/WPA keys by analyzing encrypted network traffic.
	OpenVas	Vulnerability scanning and assessment	Performs detailed scans to identify vulnerabilities, missing patches, and other weaknesses.
Bansal, Shravan, and Pawan (2021)	Nmap	Network scanning, OS fingerprinting, service detection	Identifies active hosts, open ports, services running on devices, OS details, and firewalls.
	Nessus	Vulnerability scanning and assessment	Scans for vulnerabilities in networks, misconfigurations, default passwords, DoS vulnerabilities.
	Metasploit Framework	Exploiting vulnerabilities, providing payloads, gaining control	Provides a modular approach to exploit known vulnerabilities, deliver payloads, and access systems remotely.
	Wireshark	Network traffic analysis, protocol sniffing	Analyzes network traffic, protocols used, and data exchanged between systems, enabling identification of attacks.
	Aircrack-ng	Cracking Wi-Fi networks, WEP/WPA security testing	Cracks WEP/WPA keys by analyzing encrypted network traffic.
	Cain & Abel	Password cracking, network sniffing, decoding scrambled passwords	Recovers passwords using methods like brute force, dictionary, and cracking encrypted passwords.
	Recon-ng	Web reconnaissance, OSINT (Open Source Intelligence) automation	Gathers OSINT data, automates data collection related to web applications, IP addresses, and services.
	John the Ripper	Password cracking (dictionary and brute-force attacks)	Cracks password hashes by attempting dictionary or brute-force attacks to find weak passwords.

Burp Suite is generally used for security testing of web applications, scanning for vulnerabilities, and exploiting common web application flaws. Hydra, on the other hand, is a very fast brute-forcing tool that can crack multiple protocols at once. The absence of these tools leaves an open gap in exploring web application security and password cracking techniques. In summary, Bansal, Shravan, and Pawan elaborate on sophisticated methodologies, while Mamilla focusses on foundational tools. Both, however, overlooked specialised tools like Burp Suite and Hydra, which are essential to thorough penetration testing programs like PWK[5]. A more comprehensive understanding of contemporary penetration testing is provided by combining the two methods.

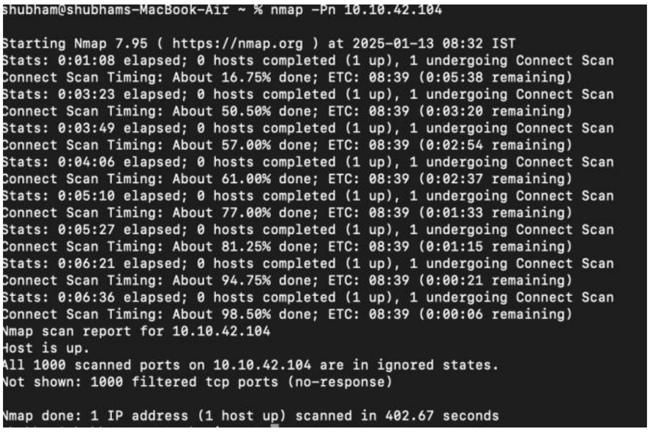
4. Implementation

A comparison of various cybersecurity tools according to their function, nature, results, and efficacy is shown in Table 5. The table illustrates the various functions that these tools carry out in penetration testing, including access-gaining, exploitation, and reconnaissance. Every tool is assessed according to how well it performs in particular scenarios, highlighting both its advantages and disadvantages.

Table 5 Comparative Analysis of Different Cybersecurity Tools Based on Their Purpose, Type, Output, And Effectiveness

Tools	phases	Type	Output	Effectiveness(%)
Nmap	Reconnaissance	Network scanning	Scanned 1000 ports on host 10.10.42.104 All 60 ports filtered(No response).Host detected as up.	60%
HashCat	Gaining access	Password hashes	Hashcat could not load hashes due to kernel issue. Missing kernel file: /Users/shubham/.cache/hashcat/kernels/shared.df4b9a60.kernel	0% (due to failure)
Metaspolit	Exploitation	Service Exploitation	199.59.243.228:21 - Port used by the backdoor bind listener is already open, but no shell detected.	40%
Burpsuit	Exploitation	SQL injection tetsing	Successfully logged in using a crafted SQL injection payload. Payload bypassed authentication by exploiting poor SQL query handling.	80%

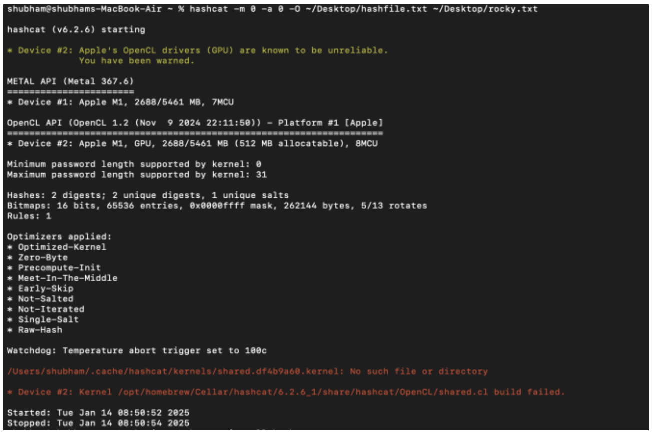
Nmap was used for reconnaissance, effectively scanning the network to identify hosts and services, with a success rate of 60%.



```
shubham@shubhams-MacBook-Air ~ % nmap -Pn 10.10.42.104

Starting Nmap 7.95 ( https://nmap.org ) at 2025-01-13 08:32 IST
Stats: 0:01:08 elapsed; 0 hosts completed (1 up), 1 undergoing Connect Scan
Connect Scan Timing: About 16.75% done; ETC: 08:39 (0:05:38 remaining)
Stats: 0:03:23 elapsed; 0 hosts completed (1 up), 1 undergoing Connect Scan
Connect Scan Timing: About 50.50% done; ETC: 08:39 (0:03:20 remaining)
Stats: 0:03:49 elapsed; 0 hosts completed (1 up), 1 undergoing Connect Scan
Connect Scan Timing: About 57.00% done; ETC: 08:39 (0:02:54 remaining)
Stats: 0:04:06 elapsed; 0 hosts completed (1 up), 1 undergoing Connect Scan
Connect Scan Timing: About 61.00% done; ETC: 08:39 (0:02:37 remaining)
Stats: 0:05:10 elapsed; 0 hosts completed (1 up), 1 undergoing Connect Scan
Connect Scan Timing: About 77.00% done; ETC: 08:39 (0:01:33 remaining)
Stats: 0:05:27 elapsed; 0 hosts completed (1 up), 1 undergoing Connect Scan
Connect Scan Timing: About 81.25% done; ETC: 08:39 (0:01:15 remaining)
Stats: 0:06:21 elapsed; 0 hosts completed (1 up), 1 undergoing Connect Scan
Connect Scan Timing: About 94.75% done; ETC: 08:39 (0:00:21 remaining)
Stats: 0:06:36 elapsed; 0 hosts completed (1 up), 1 undergoing Connect Scan
Connect Scan Timing: About 98.50% done; ETC: 08:39 (0:00:06 remaining)
Nmap scan report for 10.10.42.104
Host is up.
All 1000 scanned ports on 10.10.42.104 are in ignored states.
Not shown: 1000 filtered tcp ports (no-response)

Nmap done: 1 IP address (1 host up) scanned in 402.67 seconds
```

Figure 1 Shows the Scan Port on 10.10.24.104 ip

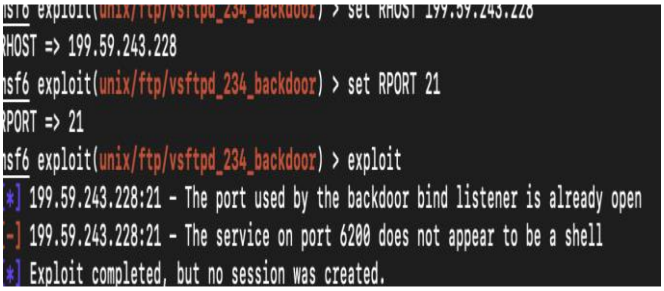
HashCat, aimed at cracking password hashes, failed due to a kernel issue, resulting in 0% effectiveness (Figure 1).



```
shubham@shubhams-MacBook-Air ~ % hashcat -m 0 -a 0 -O ~/Desktop/hashfile.txt ~/Desktop/rocky.txt

hashcat (v6.2.6) starting

* Device #2: Apple's OpenCL drivers (GPU) are known to be unreliable.
             You have been warned.

METAL API (Metal 367.6)
=======================
* Device #1: Apple M1, 2688/5461 MB, 7MCU

OpenCL API (OpenCL 1.2 (Nov  9 2024 22:11:50)) - Platform #1 [Apple]
====================================================================
* Device #2: Apple M1, GPU, 2688/5461 MB (512 MB allocatable), 8MCU

Minimum password length supported by kernel: 0
Maximum password length supported by kernel: 31

Hashes: 2 digests; 2 unique digests, 1 unique salts
Bitmaps: 16 bits, 65536 entries, 0x0000ffff mask, 262144 bytes, 5/13 rotates
Rules: 1

Optimizers applied:
* Optimized-Kernel
* Zero-Byte
* Precompute-Init
* Meet-In-The-Middle
* Early-Skip
* Not-Salted
* Not-Iterated
* Single-Salt
* Raw-Hash

Watchdog: Temperature abort trigger set to 100c

/Users/shubham/.cache/hashcat/kernels/shared.df4b9a60.kernel: No such file or directory

* Device #2: Kernel /opt/homebrew/Cellar/hashcat/6.2.6_1/share/hashcat/OpenCL/shared.cl build failed.

Started: Tue Jan 14 08:50:52 2025
Stopped: Tue Jan 14 08:50:54 2025
```

Figure 2 Shows the Result of Password Cracking Using Hashcat

Metasploit was employed for service exploitation, but it achieved limited success, with only partial exploitation and no shell obtained, resulting in 40% effectiveness (Figure 2).

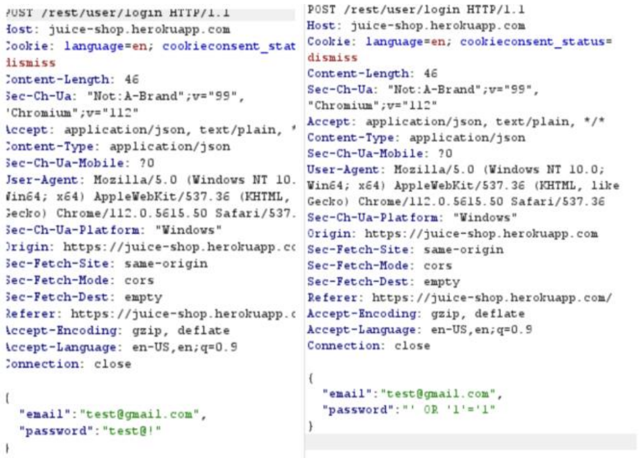


```
isf6 exploit(unix/ftp/vsftpd_234_backdoor) > set RHOST 199.59.243.228
RHOST => 199.59.243.228
isf6 exploit(unix/ftp/vsftpd_234_backdoor) > set RPORT 21
RPORT => 21
isf6 exploit(unix/ftp/vsftpd_234_backdoor) > exploit
*] 199.59.243.228:21 - The port used by the backdoor bind listener is already open
-] 199.59.243.228:21 - The service on port 6200 does not appear to be a shell
*] Exploit completed, but no session was created.
```

Figure 3 Shows the Attack Done using Metasploit on 199.52.243.228 ip

e ISSN: 2584-2854
Volume: 03
Issue:02 February 2025
Page No: 274-282

Burp Suite demonstrated a high success rate (80%) in SQL injection testing, successfully bypassing authentication using crafted payloads (Figure 3).



```
?OST /rest/user/login HTTP/1.1
Host: juice-shop.herokuapp.com
:ookie: language=en; cookieconsent_stat
lismiss
:ontent-Length: 46
Sec-Ch-Ua: "Not:A-Brand";v="99",
'Chromium";v="112"
Accept: application/json, text/plain, '
:ontent-Type: application/json
Sec-Ch-Ua-Mobile: ?0
Jser-Agent: Mozilla/5.0 (Windows NT 10.
Vin64; x64) AppleWebKit/537.36 (KHTML,
Gecko) Chrome/112.0.5615.50 Safari/537.
Sec-Ch-Ua-Platform: "Windows"
)rigin: https://juice-shop.herokuapp.cc
Sec-Fetch-Site: same-origin
Sec-Fetch-Mode: cors
Sec-Fetch-Dest: empty
Referer: https://juice-shop.herokuapp.c
Accept-Encoding: gzip, deflate
Accept-Language: en-US,en;q=0.9
:onnection: close

{
  "email":"test@gmail.com",
  "password":"test@!"
}
```

```
POST /rest/user/login HTTP/1.1
Host: juice-shop.herokuapp.com
Cookie: language=en; cookieconsent_status=
dismiss
Content-Length: 46
Sec-Ch-Ua: "Not:A-Brand";v="99",
"Chromium";v="112"
Accept: application/json, text/plain, */*
Content-Type: application/json
Sec-Ch-Ua-Mobile: ?0
User-Agent: Mozilla/5.0 (Windows NT 10.0;
Win64; x64) AppleWebKit/537.36 (KHTML, like
Gecko) Chrome/112.0.5615.50 Safari/537.36
Sec-Ch-Ua-Platform: "Windows"
Origin: https://juice-shop.herokuapp.com
Sec-Fetch-Site: same-origin
Sec-Fetch-Mode: cors
Sec-Fetch-Dest: empty
Referer: https://juice-shop.herokuapp.com/
Accept-Encoding: gzip, deflate
Accept-Language: en-US,en;q=0.9
Connection: close

{
  "email":"test@gmail.com",
  "password":"' OR '1'='1"
}
```

Figure 4 SQL Injection

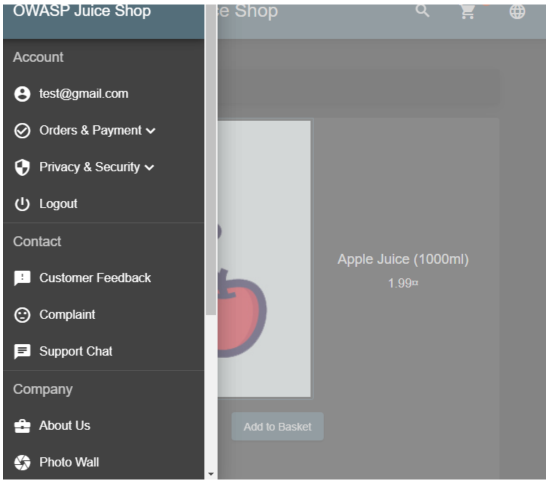


Figure 5 Shows the Successful Login After SQL Injection

5. Latest Cybersecurity Involvement in Projects

In light of emerging technologies and contemporary concerns, cybersecurity is evolving. These days, it is possible to use machine learning and artificial intelligence (AI) to predict vulnerabilities, analyse large databases, and identify suspicious activity. Threats are detected in real time with the aid of tools like Cyborg Security and Darktrace. Platforms like AWS Security Hub and Azure Security Centre keep an eye on safe APIs and multi-cloud setups for cloud security. The identity checks are more stringent at Zero Trust. The deployment of solutions like Okta and Prisma Access necessitates ongoing monitoring. In the IoT, tools like Shodan and IoT Inspector lock up smart devices and prevent firmware vulnerabilities. DevSecOps involves integrating security in software development; for instance, Snyk and Veracode are tools that automate scanning for vulnerabilities. Blockchain security includes protection of smart contracts and detection of fraud through tools like MythX and CertiK. Popular tools are penetration testing, which includes tools like Metasploit and Kali Linux; vulnerability scanners include Nessus and OpenVAS. Bug bounty platforms, such as HackerOne, and threat intelligence tools, including Recorded Future, help in unearthing security issues. Tools to tackle newer challenges include Acronis Cyber Protect for ransomware, deepfakes, and supply chain attacks, and Sonatype Nexus for managing open source. Overall, cybersecurity relies on innovative tools and practices to protect against ever-changing threats, shown in Figure 4 & Figure 5.

6. Gap Analysis

Several challenges face penetration testing in adapting to modern cybersecurity needs. Traditional approaches focus on IT systems and do not take into account emerging technologies such as IoT, blockchain, and smart cities, which thus remain vulnerable. Automated tools, including Nmap and Nessus, are efficient but miss advanced vulnerabilities, while manual testing is comprehensive but costly and impractical for many organizations. There is an underutilized balance hybrid approach. Most methodologies do not provide real-world case studies, thus being less practical. Human factors such as employee training and social engineering are often ignored, though critical. Small businesses are also at a disadvantage with the high cost of testing, hence the need for affordable solutions. Current traditional methods of testing lack the adaptability to the ever-changing environment in cloud services and CI/CD pipelines. Moreover, newer threats, like API attacks, and supply chain attacks are insufficiently addressed in the present models. Therefore, novel frameworks and approaches are needed in order to counter the latest changing technologies, evolving attack vectors, and new requirements of organizations.

7. Future Direction

Future penetration testing needs to address new challenges from new technologies such as blockchain, IoT, and smart cities, each posing unique security risks that call for specific approaches. Automated tools can speed up the process while manual methods guarantee accuracy. AI-driven tools can simulate advanced attacks, thus providing faster and more targeted tests. Affordable and open-source tools are the need of the hour for small businesses to enhance security cost-effectively. Ethical concerns, including data privacy and proper consent, must be governed by clear rules. Addressing new risks like API vulnerabilities and supply chain issues requires further research and improved tools. Real-world examples and detailed reports with actionable solutions will strengthen penetration testing and better prepare us for modern cyberattacks.

Conclusion

This literature reviewed shows how the use of penetration testing identifies and mitigates or eliminates actual cybersecurity vulnerability. Different authors mentioned different phases and types of penetration testing by applying appropriate methods according to specific strength or weakness. Indeed, founding pieces focus on pure theoretical aspects along with critical tools, while more advanced contributions include practical applications, real-world attack methods, and specialized tools, among others. What is missing in such a comprehensive effort is the full integration of practical tools, such as Burp Suite and Hydra, into the arsenal. Overall, literature points out that cybersecurity threats change in nature, testing methodologies have to alter as well, and knowledge has to be attained both theoretically and practically to fortify organizational security defenses.

References

[1]. Shravan, K., Bansal, N., & Bhadana, P. (2014). Penetration testing: A review. COMPUSOFT: An International Journal of Advanced Computer Technology, 3(4), [Volume III, Issue IV]. ISSN: 2320-0790.

[2]. Yaqoob, I., Hussain, S. A., Mamoon, S., Naseer, N., Akram, J., & ur Rehman, A. (2017). Penetration testing and vulnerability assessment. EverScience Publications, 7(8), 10. ISSN: 2395-5317.

[3]. Ami, P., & Hasan, A. (2012). Seven phrase penetration testing model. International Journal of Computer Applications, 59(5).

[4]. Mamilla, S. R. (2021). A study of penetration testing processes and tools (Master's thesis, California State University, San Bernardino). CSUSB ScholarWorks. Retrieved from https://scholarworks.lib.csusb.edu/cgi/viewcontent.cgi?article=2394&context=etd

[5]. Offensive Security. (2021). Penetration testing with Kali Linux (v2.0.1): The official training courseware for the Offensive Security Certified Professional (OSCP) exam. Offensive Security.