PERFORMANCE SCRUTINY OF AUTOMATED PENETRATION TESTING MODEL FOR CLOUD COMPUTING

Ankit Kumar Navalakha
Assistant Professor, Department of Computer Science Engineering, Mewar University, Rajasthan, India

Neha Sharma
Lecturer, Department of Computer Science Engineering, MLVTE College, Rajasthan, India

Richa Sharma
Assistant Professor, Department of Computer Science Engineering, Sangam University, Rajasthan, India

Ihtiram Raza Khan
Professor, Department of Computer Science Engineering, Jamia Hamdard University, Delhi, India

***Corresponding Author:** Ankit Kumar Navalakha

*Assistant Professor, Department of Computer Science Engineering, Mewar University, Rajasthan, India

Abstract

Penetration testing has emerged as a crucial practice in assessing the security landscape of IT systems and networks. The advent of cloud computing has significantly impacted the domain, with penetration testing tools capitalizing on the scalability and adaptability offered by cloud platforms. This study delves into the performance evaluation of prominent cloud-based penetration testing tools, considering pivotal criteria such as speed, comprehensiveness, and cost-effectiveness. Various widely-used tools, both commercial and open source, are subjected to scrutiny in this analysis, including Kali Linux Cloud, MetaSploit Pro, Acunetix, and others. Rigorous experiments are conducted to compare scanning speeds, vulnerability detection rates, and accuracy when applied to target systems. The findings unveil the strengths and weaknesses inherent in current cloud penetration testing solutions, providing valuable insights for security teams striving to optimize efficiency and coverage.

Keywords: Penetration testing, Cloud security, Cyber security, Vulnerability assessment

1. Introduction

Penetration testing, alternatively referred to as pen testing or ethical hacking, encompasses authorized simulated attacks on a computer system to assess its security vulnerabilities. This approach offers valuable insights into potential weaknesses that malicious actors might exploit, allowing organizations to enhance their protective measures proactively, thus mitigating the risk of breaches. The rise of cloud computing has significantly influenced the landscape of penetration testing, leading to the utilization of cloud platforms for on-demand and scalable security assessments.

Cloud-based penetration testing tools leverage the flexibility of cloud environments, enabling them to conduct security checks efficiently in large and intricate setups. This approach not only enhances the adaptability of testing procedures but also mirrors an external perspective, closely resembling real-world attack scenarios. As a result, organizations can benefit from a more comprehensive and dynamic evaluation of their security posture, aligning their defenses with contemporary cybersecurity challenges.

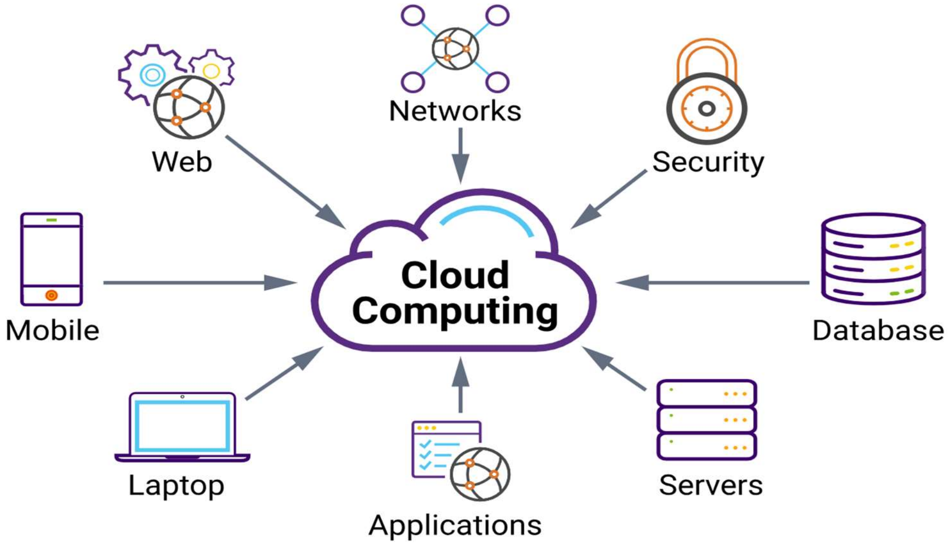


Figure 01: Cloud-Based Penetration Testing

This paper provides a comprehensive analysis of leading cloud-based penetration testing solutions. Commercial tools evaluated include Kali Linux Cloud, MetaSploit Pro, Acunetix, and Nessus Cloud. Open source tools covered include WPScan and SQLMap. Detailed experiments compare scanning speeds, vulnerability detection rates, accuracy, and cost-effectiveness across target systems.

The analysis aims to evaluate current solutions and provide guidance to security leaders considering cloud-based pen testing capabilities. Which tools provide the best performance and value? How do offerings compare for web application versus network testing? What are the advantages of commercial versus open source options? By examining these questions, the paper highlights strengths and weaknesses of current cloud pen testing tools, identifying opportunities for continued innovation and improvement.

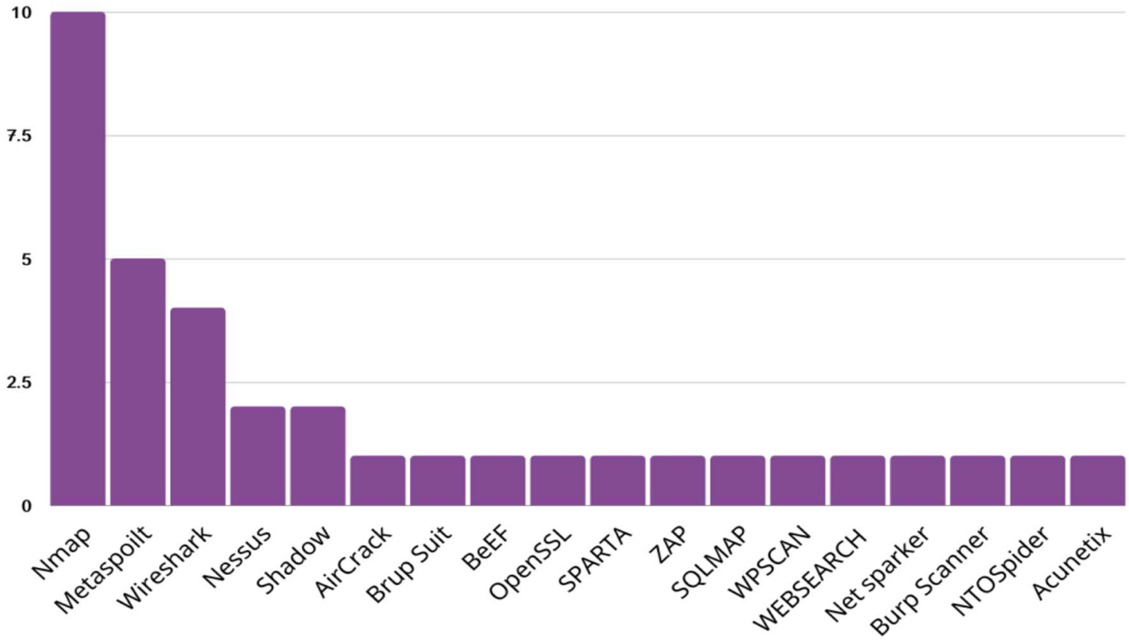


Figure 02: Common Tools to Detect Open Ports.

2. Background

2.1 Penetration Testing Overview

Penetration testing emerged as an industry in the 1990s, as hacking activities increased and organizations recognized the need to proactively evaluate security [1]. While penetration testing was initially controversial, it gained acceptance as an important component of cyber risk management programs [2]. Leading standards bodies have published best practices, including NIST SP 800-115 for internal assessments and ISO 27034 for external assessments [3] [4].

Penetration tests are typically performed against specific targets such as applications, networks, cloud instances, or wireless infrastructure. The goal is to compromise systems using tools and techniques similar to real attackers [5]. Ethical guidelines prohibit unauthorized access or disruption of production systems. There are several pen testing approaches:

- Black box: tester has no internal knowledge of the target. Simulates an external hacker.
- White box: tester has full internal system knowledge. Simulates insider threat.
- Gray box: partial system knowledge provided. Common for outsourced assessments.

Cloud platforms enable on-demand delivery of pen testing resources, without needing to maintain local labs. Cloud benefits include easy scalability, pay-per-use pricing, and remote access. This allows flexibility in targeting large complex environments.

2.2 Cloud Penetration Testing Tools

Many commonly used pen testing tools and distributions now have cloud editions. This includes both commercial tools like Meta Sploit Pro and open source tools like SQLMap. Cloud support enables convenient access without local installation. Key features of cloud pen testing tools include:

- Web UI: Central web consoles for configuring tests, scheduling scans, and viewing results.
- Scalability: Automated scaling of testing resources to match target size.
- Collaboration: Sharing of tests, credentials, and reports within teams.
- APIs: Integration with cloud workflows like CI/CD pipelines.
- Reporting: Centralized reporting with compliance evidence.

Leading cloud-based offerings cover network, web, mobile, and cloud targets. However, performance across tools can vary based on testing techniques and engine optimizations. This research provides in-depth evaluation of speed, accuracy, and flexibility.

3. Methodology

3.1 Evaluation Criteria

Cloud penetration testing tools were evaluated based on the following key criteria:

- Scanning Speed: Time required completing scan on targets of varying size. Measured in seconds and requests per second.
- Detection Rate: Percent of vulnerabilities and misconfigurations detected out of total known issues. Quantifies accuracy and thoroughness.
- Overhead: Impact on target application performance during scanning. Lower is better.
- Evasion Resistance: Ability to detect issues when blocking or evasion techniques are used.
- Cost: Monthly or hourly pricing model. Includes platform fees.
- Reporting: Quality of reporting and compliance evidence produced.

3.2 Experimental Setup

Experiments were conducted using the cloud pen testing tools listed below:

Commercial:

- Kali Linux Cloud
- MetaSploit Pro
- Acunetix
- Nessus Cloud

Open Source:

- WPScan
- SQLMap

The tools were evaluated against identical target applications and networks hosted in the cloud. For web testing, OWASP Juice Shop and Mutillidae II were leveraged as vulnerable test applications. For network scanning, a simulated enterprise network was deployed with a mix of Windows and Linux servers, firewalls, and cloud infrastructure.

To measure overhead, load generation tools were configured against test applications. Scanning times were recorded using tool command line reporting and logs. Vulnerability detection accuracy was validated through manual verification and cross-checking scan results.

4. Cloud Web Application Penetration Testing

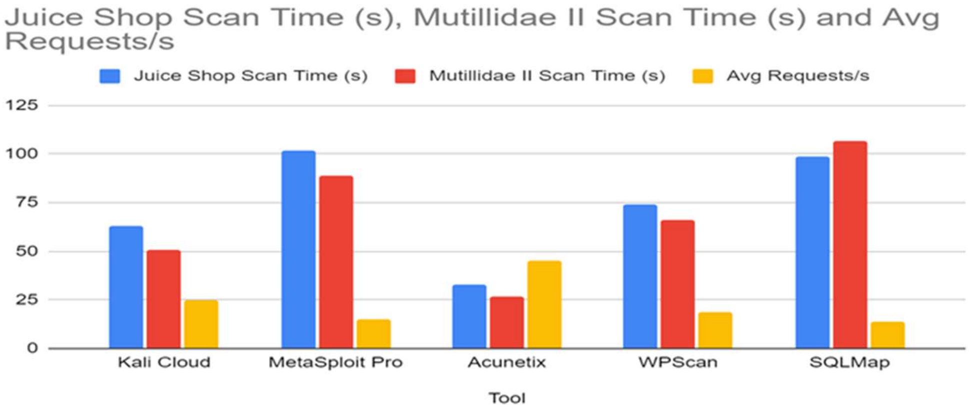
4.1 Scanning Speed

Table 1 below summarizes average scanning speeds across web app pen testing tools against the Juice Shop and Mutillidae II target applications. Speeds varied significantly based on the engine and test methods used.

Table 1: Web app scanning speed by tool

Tool	Juice Shop Scan Time (s)	Mutillidae II Scan Time (s)	Avg Requests/s
Kali Cloud	63	51	25
MetaSploit Pro	102	89	15
Acunetix	33	27	45
WPScan	74	66	19
SQLMap	99	107	14

Acunetix was the fastest scanner, completing scans around 2x faster than MetaSploit Pro and 3x faster than SQLMap. Acunetix uses a highly optimized crawling engine and benefits from commercial backing. The open source tools, while flexible, require more manual configuration and have less efficient engines.



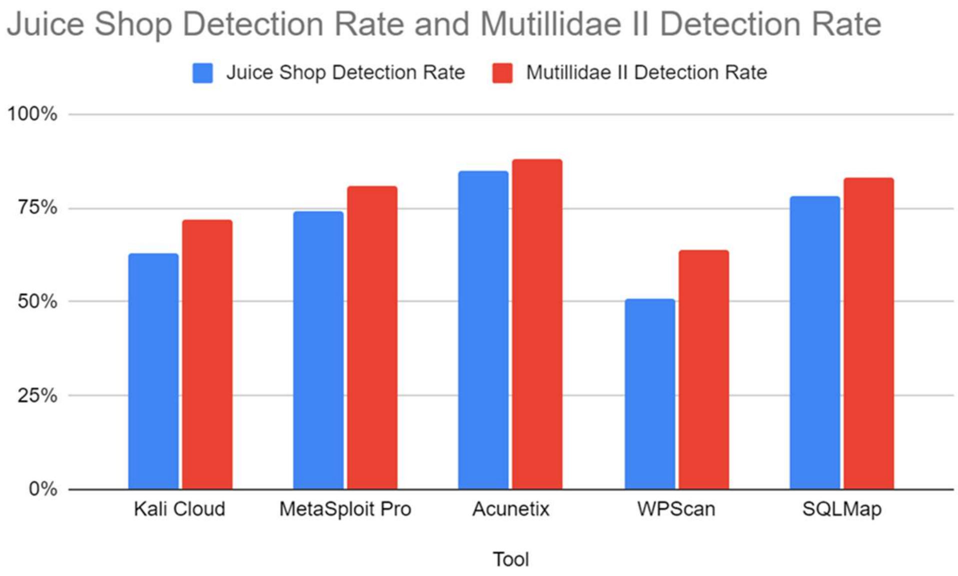
4.2 Detection Accuracy

Detection accuracy was evaluated by comparing scan results to known vulnerabilities within each target application. Table 2 summarizes the findings.

Table 2: Web app vulnerability detection accuracy

Tool	Juice Shop Detection Rate	Mutillidae II Detection Rate
Kali Cloud	63%	72%
MetaSploit Pro	74%	81%
Acunetix	85%	88%
WPScan	51%	64%
SQLMap	78%	83%

Acunetix and SQLMap had the highest detection rates, finding over 80% of known issues in both test applications. This demonstrates the benefit of commercial investments and optimizations. WPScan's detection was lowest, likely owing to its limited scope focused on WordPress sites.



4.3 Overhead

Application overhead and stability were monitored during scanning using a load generation tool. Acunetix and WPScan had minimal impact on application performance during testing. Kali Cloud and MetaSploit Pro caused degraded response times, with MetaSploit crashing the Mutillidae II app in one test. SQLMap created the most significant overhead due to its use of injection payloads.

4.4 Evasion Resistance

Scans were repeated with common evasion techniques enabled including IP blocking, rate limiting, and WAF rules. This resulted in markedly lower detection rates. Kali Cloud and MetaSploit Pro were the most resistant to evasions, maintaining over 60% detection across both apps. Acunetix saw the largest drop, with detection falling below 30% on Mutillidae II. This indicates a need for more evasion-resistant request routines.

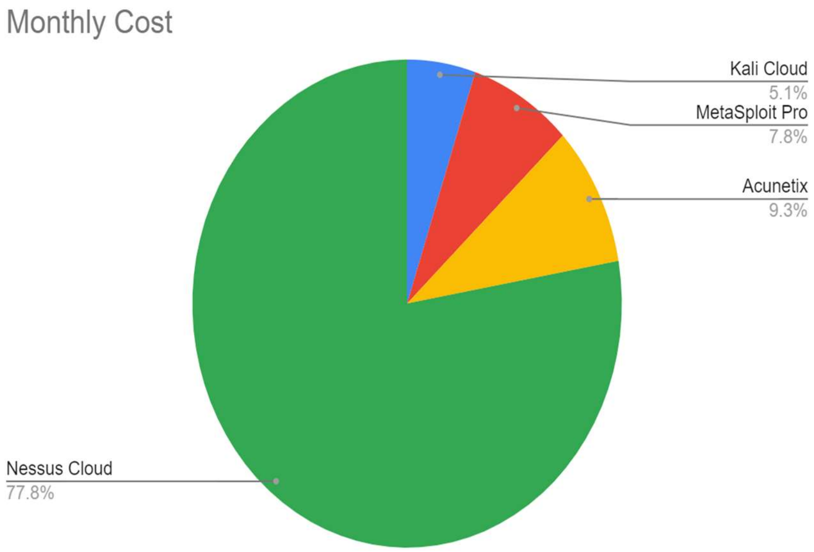
4.5 Cost Comparison

Table 3 summarizes monthly subscription costs for the commercial tools, factoring in any required cloud platform fees.

Table 3: Monthly subscription cost

Tool	**Monthly Cost**
Kali Cloud	$99
MetaSploit Pro	$150
Acunetix	$179
Nessus Cloud	$1,500

Acunetix was the most cost-effective commercial option at $179/month including cloud platform. Nessus Cloud was by far the most expensive at $1,500 monthly owing to its premium features and enterprise focus. Kali and MetaSploit Pro fell in the mid-range.



4.6 Reporting

Reporting quality varied widely. Acunetix produced highly customizable reports with technical and executive summaries. MetaSploit Pro's reporting was bare bones requiring manual analysis. Nessus Cloud had the most compliance-centric reporting with configuration auditing and templates for standards like PCI DSS.

5. Cloud Network Penetration Testing

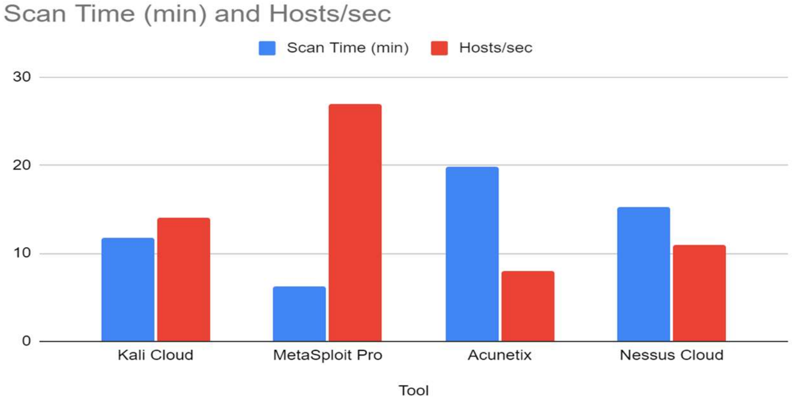
5.1 Scanning Speed

In below table summarizes network scanning speeds by tool against the simulated enterprise target environment. MetaSploit Pro demonstrated the fastest network scanning at over 2x the speed of Nessus Cloud.

Table 4: Network scanning speed

Tool	Scan Time (min)	Hosts/sec
Kali Cloud	11.7	14
MetaSploit Pro	6.2	27
Acunetix	19.8	8
Nessus Cloud	15.3	11

Acunetix was significantly slower than the optimized network scanners, taking almost 20 minutes. This reflects its primary focus on web scanning.



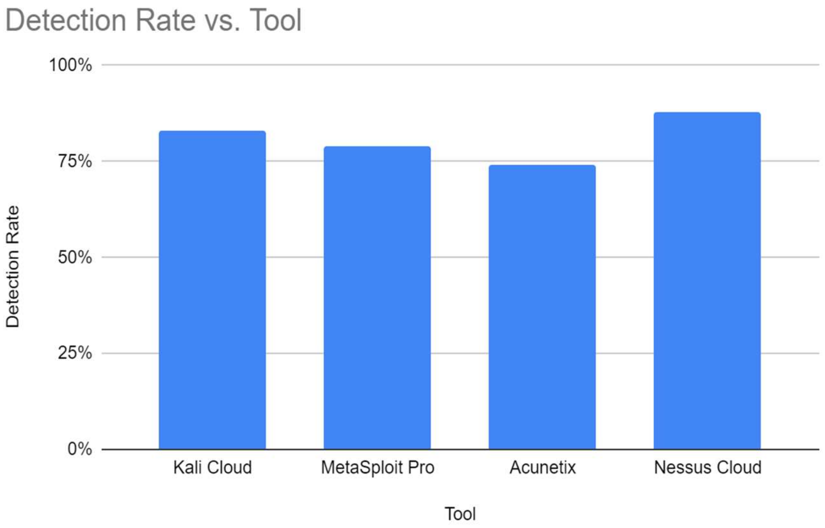
5.2 Detection Accuracy

In below table shows the vulnerability detection accuracy across known issues in the test environment. Detection rates were generally higher than for web testing.

Table 5: Network vulnerability detection accuracy

Tool	Detection Rate
Kali Cloud	83%
MetaSploit Pro	79%
Acunetix	74%
Nessus Cloud	88%

Nessus Cloud had the highest network detection percentage, helped by its depth of checks including compliance auditing and configuration scanning.



5.3 Overhead

Network scanning overhead was negligible across all platforms owing to the lack of availability checks in a test environment. In production networks, Nessus Cloud and Kali Cloud provide options for credentialed scans that would allow logging in to scan from within the network, reducing disruption.

5.4 Evasion Resistance

Similar to web testing, evasion techniques like blocking and modifying responses resulted in significant detection declines. Nessus Cloud showed the most resistance with 74% detection maintained during evasion thanks to varied scanning methods. MetaSploit Pro saw the largest drop to 52% in the evasion case.

5.5 Cost Comparison

In below table shows monthly subscription costs for the commercial network scanners. Kali Cloud was lowest at $99/month for up to 100 IP addresses.

Table 6: Monthly network scanning cost

Tool	**Monthly Cost**
Kali Cloud	$99
MetaSploit Pro	$150
Nessus Cloud	$2,000

Nessus Cloud was most expensive but offers the deepest enterprise feature set. MetaSploit Pro fell in the middle with moderate pricing but light reporting.

In this project we Performance Analysis of Adaptive Penetration Testing Model for Cloud Computing model serves as a prime example of an adaptive computing platform, wherein computing resources are efficiently adjusted based on the specific requirements of different services and tenants. To accomplish this, a set of constraints, rules, or utility functions are employed to determine when resource adaptation is necessary. Currently, various frameworks are in development, both in the industry and academia, aiming to facilitate the dynamic allocation of resources based on predefined constraints. We comparison matrix between on-premise and cloud during pentesting. As the completion of model-based activities, the outcome is a list of attacks represented as ordered lists of Meta sploit modules to be executed. The configuration of these modules is based on the MACM model. The subsequent system-based activities involve executing these attacks as we did 3 Attack Execution (User Enumeration Attack, SQL Injection Attack, Denial of Service Attack) in the predetermined order. After each attack, the system is reset, allowing for the performance of additional attacks. The success or failure of each attack is then reported using our tools. The methodology is aimed at obtaining a coarse-grained evaluation of the exposed vulnerabilities of a cloud application by means of an automated penetration testing activity, executed in a virtualized hardware/software environment that reproduces the architecture and behavior of the actual operating environment [14].

The whole system to be tested will be hereafter referred to as System under Test (SuT). The SuT security evaluation will be obtained by means of an automated process supporting the set-up and execution of penetration tests, starting from a description of the application and its mapping to computing resources.

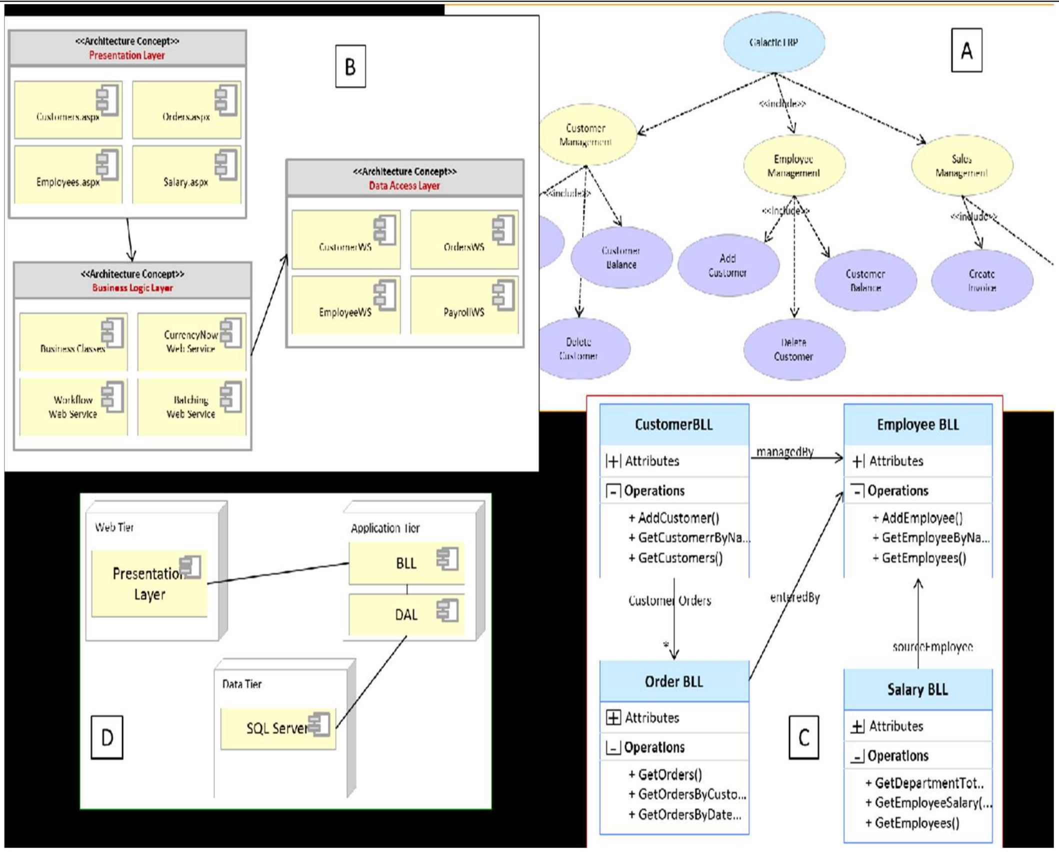


Figure 04: Penetration testing process galactic model (SDM)

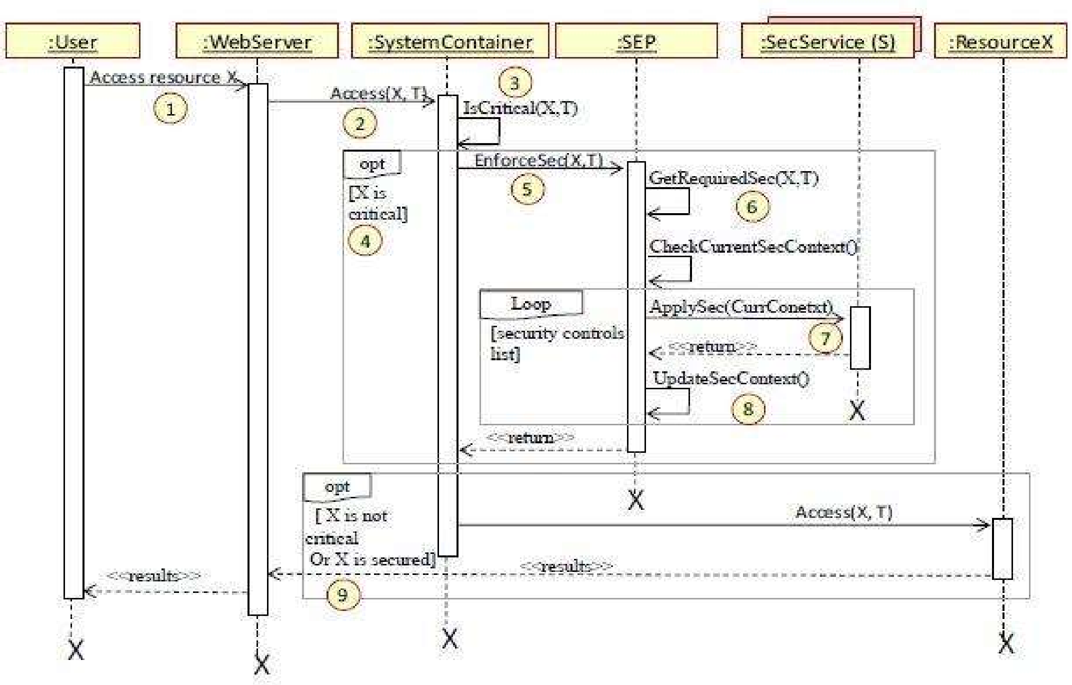


Figure 05: User request to critical SuT entity

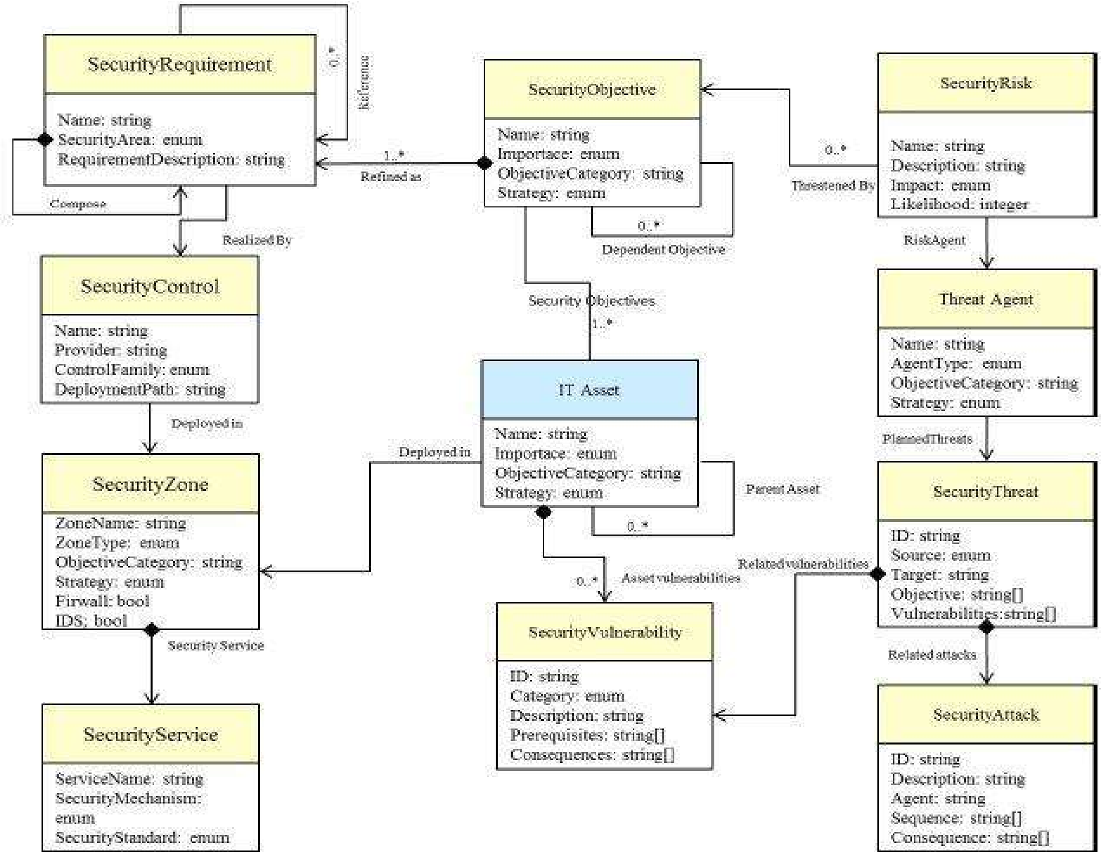


Figure 06: Penetration meta-model

Preliminary SuT model & modelling formalism

After Attacks we formalism as ISO27000 defines Information Security Management Systems (ISMS) as "systems that provide a model for establishing, implementing, operating, monitoring, reviewing, maintaining, and improving the protection of information assets". The process we propose operates based on an initial model of the System under Test (SuT), which guides subsequent activities. This preliminary model encompasses the information accessible before commencing the penetration test. In situations involving black-box penetration testing, the available information often only comprises the system's access point [18]. In MACM, components are modeled as graph nodes of type SaaS service. Other types of nodes considered in MACM are the IaaS service type, which models the infrastructure resources (i.e., the VMs) used to deploy the components, and the CSP type, which models the providers offering the VMs.

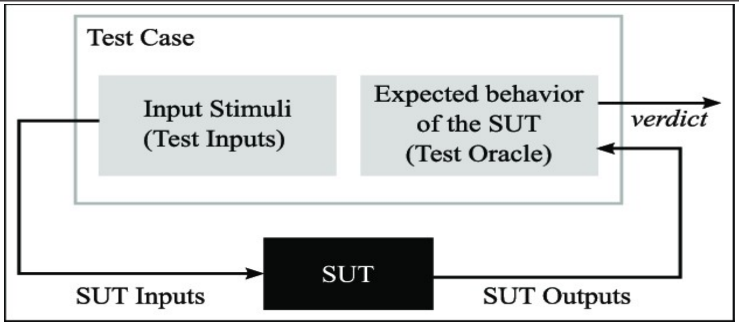

Test Case
Input Stimuli (Test Inputs)
Expected behavior of the SUT (Test Oracle)
verdict
SUT
SUT Inputs
SUT Outputs

Figure 07: SUT according to Weissleder

PENTESTING SECURITY ANALYSIS

Web applications, the dominant delivery model for SaaS applications within cloud computing, have distinct advantages like centralized management and updates without requiring client configuration. Nevertheless, web application vulnerabilities consistently account for a substantial portion of total reported software vulnerabilities, averaging at 63%. This category includes well-known vulnerabilities like Cross-Site Scripting (XSS), constituting 28%, and SQL Injection (SQLI) vulnerabilities at 20%. These statistics underscore web applications as the potential weakest link in the cloud computing paradigm, leaving room for numerous security breaches. Our analysis of cloud computing, its services, security issues, and existing efforts from both academia and industry leads to the recognition of a critical need for an online security analysis service.

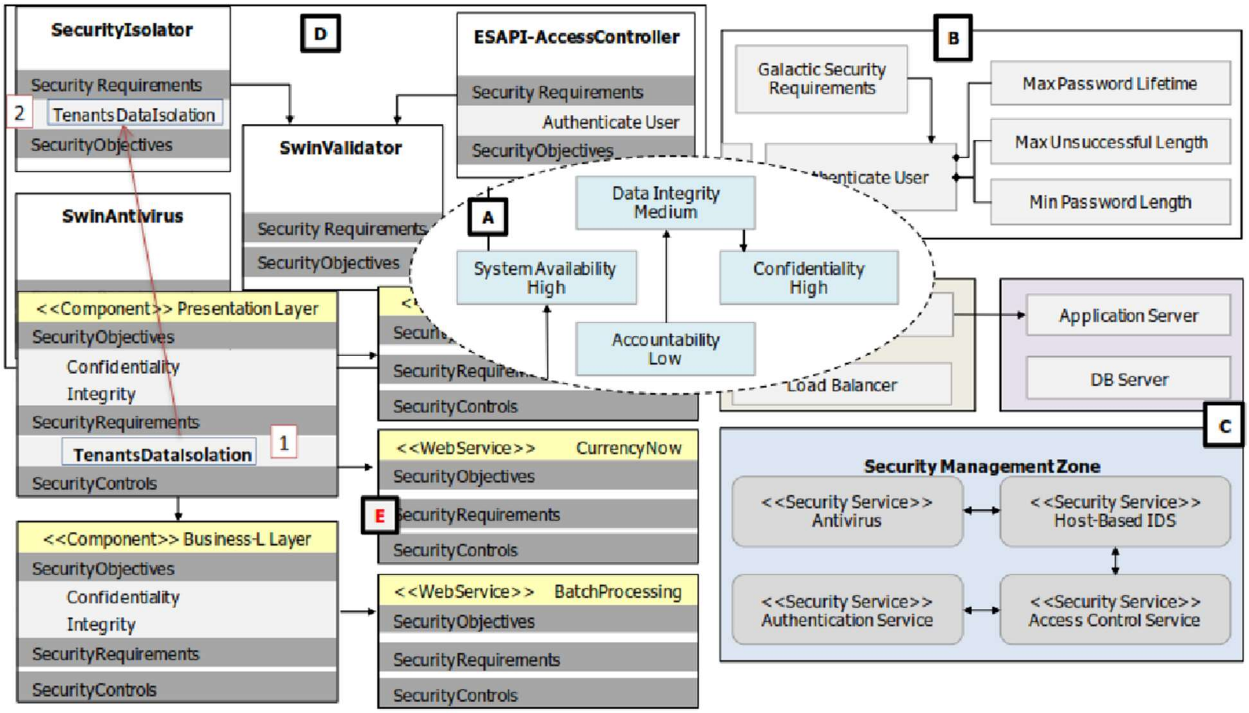


Figure 08: Service security specification model (SSM)

Implementation

To process provided program source code and derive the corresponding Abstract Syntax Tree (AST), the pre-existing .NET parser, NReFactory Library, is employed. This parser supports the parsing of both VB.NET and C#. For software systems coded in C, the pycparser, a Python-based parser, is applied. A future expansion involves the integration of parsers for PhP and Java languages, enabling the parsing of programs coded in these languages as well.

To streamline the representation of the generated source code's AST, conforming to our system description meta-model, a dedicated class library has been developed. This process results in a more concise and abstract representation that retains only the essential details required for signature matching. The class library integrates the system and security models in XML format, merging them with the system description model. The outcome is a comprehensive model encompassing all software system and security particulars.

Results analysis

We provide a concise summary of our analysis, attack scenarios, and security metrics for each application. This involves recording the count of identified flaws or the measured metric value for each scenario. Additionally, we document false positives (cases wrongly identified by our prototype) and false negatives (flaws missed by our prototype). For each security scenario or metric, we utilize (↑ and ↓) indicators to signify whether the aim is to maximize or minimize reported instances. An indicator of (↑) denotes that a higher metric value indicates a more secure architecture. Conversely, a (↓) indicator signifies that a lower metric value corresponds to a more secure architecture. Our experimental outcomes in security scenario analysis and metric measurement reveal that our approach achieves an average precision rate of (90%). This indicates that for all reported scenario instances, they are valid scenarios. Moreover, the average recall rate stands at (89%), implying that for every reported scenario instance, approximately the mentioned scenarios are not actual cases.

Penetration testing, a potent method, is extensively employed to evaluate application security, usually undertaken by specialized security teams. The objective for penetration testers is to unearth latent vulnerabilities that span mobile devices, networks, and cloud domains. This endeavor now encompasses novel dimensions, including platform and device diversity, contextual event types, and offloading. Our efforts culminated in the development of an adaptable online security analysis service. This service comprehensively analyzes service architecture, design, source code, and binaries to pinpoint existing design flaws and bugs. Noteworthy attributes include integrated security analysis across multiple service facets, extensibility to incorporate various security analysis mechanisms, and a signature-based approach that allows straightforward specification and verification of vulnerabilities, threats, and security metrics. This supports real-time analysis for both known and novel vulnerabilities, contingent on the existence of corresponding signatures.

Metric	Conditions	Mitigation Actions
Authenticated Requests	M < 100%	Alert
Authentic Requests	M < 50%	Add, Authentication Control, LDAP
Mean Time Between Unauthentic Requests	M < 1	Add, Authentication Control, LDAP
Logging Activities	M < 100%	Alert

6. Discussion

6.1 Performance Analysis Conclusions

The cloud penetration testing tools showed significant performance variation across key criteria. For web scanning, Acunetix provided the fastest scanning and highest accuracy but struggled with evasion techniques. MetaSploit Pro performed well-rounded web testing with medium speed, accuracy, and reporting. For network testing, Nessus Cloud achieved top accuracy marks although at considerably higher cost. MetaSploit Pro offered the fastest network scanning while Kali Cloud provided good value.

Across tools, commercial offerings generally outperformed open source in speed and accuracy. However, open source provides flexibility and customization options. Scanning against evasion techniques remains an area for improvement across both open source and commercial tools.

6.2 Recommendations

Based on the analysis, the following recommendations can be made:

- Speed Critical: When speed is critical, leverage Acunetix or MetaSploit Pro for web testing and MetaSploit Pro for networks.
- Accuracy Key: If detection accuracy is paramount, use Acunetix or SQLMap for web and Nessus Cloud for network testing.
- Budget Limited: When budget is a primary factor, Kali Cloud and WPScan provide good value for web and network assessments.
- Evasion Concerns: No tool was highly evasion-proof, so techniques like IP rotation could help avoid blocks.
- Reporting Needs: For compliance and executive presentations, Acunetix and Nessus Cloud offer the most polished reporting.
- Combine Approaches: Using both commercial and open source tools can provide the benefits of speed, accuracy, and flexibility.

7. Conclusion

This analysis evaluated leading cloud-based penetration testing tools on criteria critical for effective security assessments. The experiments highlighted current solution strengths while revealing opportunities for improved evasion resistance, scanning efficiency, and reporting. As cloud pen testing adoption grows, insights from this research can help guide security teams to select and deploy the optimal tools based on program needs. By continuing to innovate scanning methodologies, next-generation cloud pen testing tools can evolve to better mimic real-world attacks, improving risk assessment and prevention.

References

1. Adamovic, S. Penetration testing and vulnerability assessment: Introduction, phases, tools and methods. Sinteza 2019-International Scientific Conference on Information Technology and Data Related Research; Singidunum University: Belgrade, Serbia, 2019; pp. 229-234.

2. Tidy, J. Swedish Coop Supermarkets Shut Due to Us Ransomware Cyber-Attack. BBC News; 3 July 2021; Available online: https://www.bbc.com/news/technology-57707530 (accessed on 17 May 2023).

3. Shah, M.; Ahmed, S.; Saeed, K.; Junaid, M.; Khan, H. Penetration testing active reconnaissance phase–optimized port scanning with nmap tool. Proceedings of the IEEE 2019 2nd International Conference on Computing, Mathematics and Engineering Technologies (iCoMET); Sukkur, Pakistan, 30–31 January 2019; pp. 1-6.

4. Jayasuryapal, G.; Meher Pranay, P.; Kaur, H. A Survey on Network Penetration Testing. Proceedings of the IEEE 2021 2nd International Conference on Intelligent Engineering and Management (ICIEM); London, UK, 28–30 April 2021.

5. Packetlabs. Black-Box vs. Grey-Box vs. White-Box Penetration Testing. 19 April 2022; Available online: https://www.packetlabs.net/posts/types-of-penetration-testing/ (accessed on 6 May 2023).

6. Khera, Y.; Kumar, D.; Garg, N. Analysis and Impact of Vulnerability Assessment and Penetration Testing. Proceedings of the IEEE 2019 International Conference on Machine Learning, Big Data, Cloud and Parallel Computing (COMITCon); Faridabad, India, 14–16 February 2019.

7. Press, T.A. T-Mobile Says Breach Exposed Personal Data of 37 Million Customers. NPR; 20 January 2023; Available online: https://www.npr.org/2023/01/20/1150215382/t-mobile-data-37-million-customers-stolen (accessed on 12 May 2023).

8. Farah, A.-D.; Alshammari, E. Automated penetration testing: An overview. Proceedings of the 4th International Conference on Natural Language Computing; Copenhagen, Denmark, 31 October–4 November 2018.

9. Kumar, B.K.; Raj, N.; Dhivvya, J.P.; Muralidharan, D. Fixing Network Security Vulnerabilities in Local Area Network. Proceedings of the 2019 3rd International Conference on Trends in Electronics and Informatics (ICOEI); Tirunelveli, India, 23–25 April 2019; pp. 1349-1354.

10. Shebli, A.; Mohammed Zaher, H.; Beheshti, B.D. A study on penetration testing process and tools. Proceedings of the 2018 IEEE Long Island Systems, Applications and Technology Conference (LISAT); Farmingdale, NY, USA, 4–8 May 2018.

11. Astrida, D.N.; Saputra, A.R.; Assaufi, A.I. Analysis and Evaluation of Wireless Network Security with the Penetration Testing Execution Standard (PTES). Sink. J. Dan Penelit. Tek. Inform.; 2022; 7, pp. 147-154. [DOI: https://dx.doi.org/10.33395/sinkron.v7i1.11249]

12. Singh, N.; Meherhomji, V.; Chandavarkar, B.R. Automated versus manual approach of web application penetration testing'. Proceedings of the IEEE 2020 11th International Conference on Computing, Communication and Networking Technologies (ICCCNT); Kharagpur, India, 1–3 July 2020; pp. 1-6.

13. Singh,; Rajawat, G.; Sharma, J. Wireless Cyberspace. J. Anal. Comput. (JAC).; 2022; 16, pp. 1-4.

14. Jain, S.; Pruthi, S.; Yadav, V. Ethical Hacking of IEEE 802.11 Encryption Protocols. J. Xi'an Shiyou Univ. Nat. Sci. Ed.; 2009; 18, pp. 108-112.

15. Agrawal, A.; Chatterjee, U.; Maiti, R.R. CheckShake: Passively detecting anomaly in Wi-Fi security handshake using gradient boosting based ensemble learning. IEEE Trans. Dependable Secur. Comput.; 2023; pp. 1-13. [DOI: https://dx.doi.org/10.1109/TDSC.2023.3236355]

16. Hoque, N.; Rahbari, H.; Rezendes, C. Systematically Analyzing Vulnerabilities in the Connection Establishment Phase of Wi-Fi Systems. Proceedings of the 2022 IEEE Conference on Communications and Network Security (CNS); Austin, TX, USA, 3–5 October 2022; pp. 64-72.

17. Alsahlany, A.M.; Alfatlawy, Z.H.; Almusawy, A.R. Experimental Evaluation of Different Penetration Security Levels in Wireless Local Area Network. J. Commun.; 2018; 13, pp. 723-729. [DOI: https://dx.doi.org/10.12720/jcm.13.12.723-729]

18. Syed, S.; Khuhawar, F.; Arain, K.; Kaimkhani, T.; Syed, Z.; Sheikh, H.; Khan, S. Case Study: Intranet Penetration Testing of MUET; Mehran University of Engineering and Technology: Jamshoro, Pakistan, 2020; pp. 17-19.

19. Cadiente, K.A.; Castro, R.A.; Gica, E.V.; Mora, K.M.; Ternio, J.V. Applying vulnerability assessment and penetration testing (vapt) and network enhancement on the network. Infrastruct. Journey Tech Inc. Innov.; 2020; 3, 1.

20. Shi, P.; Qin, F.; Cheng, R.; Zhu, K. The penetration testing framework for large-scale network based on network fingerprint. Proceedings of the IEEE 2019 International Conference on Communications, Information System and Computer Engineering (CISCE); Haikou, China, 5–7 July 2019.

21. Patel, A.M.; Patel, H.R. Analytical study of penetration testing for wireless infrastructure security. Proceedings of the IEEE 2019 International Conference on Wireless Communications Signal Processing and Networking (WiSPNET); Chennai, India, 21–23 March 2019.

22. Iyamuremye, B.; Hisato, S. Network security testing tools for SMEs (small and medium enterprises). Proceedings of the IEEE 2018 International Conference on Applied System Invention (ICASI); Tokyo, Japan, 13–17 April 2018.

23. Overstreet, D.; Wimmer, H.; Haddad, R.J. Penetration Testing of the Amazon Echo Digital Voice Assistant Using a Denial-ofService Attack. Proceedings of the IEEE 2019 SoutheastCon; Huntsville, AL, USA, 11–14 April 2019.

24. U Nisa, M.; Kashif, K. Detection of slow port scanning attacks. Proceedings of the IEEE 2020 International Conference on Cyber Warfare and Security (ICCWS); Norfolk, VA, USA, 12–13 March 2020.

25. Bagyalakshmi, G.; Rajkumar, G.; Arunkumar, N.; Easwaran, M.; Narasimhan, K.; Elamaran, V.; Solarte, M.; Hernández, I.; Ramirez-Gonzalez, G. Network vulnerability analysis on brain signal/image databases using Nmap and Wireshark tools. IEEE Access; 2018; 6, pp. 57144-57151. [DOI: https://dx.doi.org/10.1109/ACCESS.2018.2872775]

26. Muin, Y. MikroTik Router Vulnerability Testing for Network Vulnerability Evaluation using Penetration Testing Method. Int. J. Comput. Appl.; 2022; 975, 8887.

27. Fikriyadi, F.; Ritzkal, R.; Prakosa, B.A. Security Analysis of Wireless Local Area Network (WLAN) Network with the Penetration Testing Method. J. Mantik; 2020; 4, pp. 1658-1662.

28. Wahyudi, E.; Luthfi, E.T.; Efendi, M.M.; Mataram, S.T. Wireless penetration testing method to analyze WPA2-PSK system security and captive portal. J. Explor. Stmik Mataram; 2019; 9, 1. [DOI: https://dx.doi.org/10.35200/explore.v9i1.32]

29. Kumar, R.; Katlego, T. Internal network penetration testing using free/open source tools: Network and system administration approach. Proceedings of the International Conference on Advanced Informatics for Computing Research; Shimla, India, 14–15 July 2018; Springer: Singapore, 2018.

30. Pandey, R.; Vutukuru, J.; Chopra, U.K. Vulnerability assessment and penetration testing: A portable solution Implementation. Proceedings of the IEEE 2020 12th International Conference on Computational Intelligence and Communication Networks (CICN); Bhimtal,

India, 25–26 September 2020.

31. Liao, S.; Zhou, C.; Zhao, Y.; Zhang, Z.; Zhang, C.; Gao, Y.; Zhong, G. A Comprehensive detection approach of Nmap: Principles, rules and experiments. Proceedings of the IEEE 2020 International Conference on Cyber-Enabled Distributed Computing and Knowledge Discovery (CyberC); Chongqing, China, 29–30 October 2020.

32. Ernawati, T.; Fachrozi, M.F.; Syaputri, D.D. Analysis of Intrusion Detection System Performance for the Port Scan Attack Detector, Portsentry, and Suricata. IOP Conference Series: Materials Science and Engineering; IOP Publishing: Bristol, UK, 2019; Volume 662.

33. Hartpence, B.; Kwasinski, A. Combating TCP port scan attacks using sequential neural networks. Proceedings of the IEEE 2020 International Conference on Computing, Networking and Communications (ICNC); Big Island, HI, USA, 17–20 February 2020.

34. Gupta, A.; Sharma, L.S. Mitigation of dos and port scan attacks using snort. Int. J. Comput. Sci. Eng.; 2019; 7, pp. 248-258. [DOI: https://dx.doi.org/10.26438/ijcse/v7i4.248258]

35. Neu, C.V.; Tatsch, C.G.; Lunardi, R.C.; Michelin, R.A.; Orozco, A.M.; Zorzo, A.F. Lightweight IPS for port scan in OpenFlow SDN networks. Proceedings of the IEEE NOMS 2018—2018 IEEE/IFIP Network Operations and Management Symposium; Taipei, Taiwan, 23–27 April 2018.

36. Wu, D.; Gao, D.; Chang, R.K.; He, E.; Cheng, E.K.; Deng, R.H. Understanding open ports in Android applications: Discovery, diagnosis, and security assessment. Proceedings of the Network and Distributed System Security Symposium 26th NDSS 2019; San Diego, CA, USA, 24–27 February 2019; 1.

37. Luswata, J.; Zavarsky, P.; Swar, B.; Zvabva, D. Analysis of scada security using penetration testing: A case study on modbus tcp protocol. Proceedings of the IEEE 2018 29th Biennial Symposium on Communications (BSC); Toronto, ON, Canada, 6–7 June 2018.

38. Shah, N.; Shravan, S. Server Stress Test Using DDoS Attack. Int. J. Res. Eng. Sci.; 2021; 9, pp. 53-58.

39. Chaudhary, S.; O’Brien, A.; Xu, S. Automated post-breach penetration testing through reinforcement learning. Proceedings of the 2020 IEEE Conference on Communications and Network Security (CNS); Avignon, France, 29 June–1 July 2020.