iJRASET
International Journal For Research in Applied Science and Engineering Technology

INTERNATIONAL JOURNAL FOR RESEARCH

IN APPLIED SCIENCE & ENGINEERING TECHNOLOGY

Volume: 13 **Issue:** VIII **Month of publication:** August 2025

DOI: https://doi.org/10.22214/ijraset.2025.73701

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 VIII Aug 2025- Available at www.ijraset.com

Adaptive Cyber Defense: Realistic SOC Workflow Implementation

Likhith G[1], Swetha P M[2]
[1]*PG Scholar,* [2]*Assistant Professor, Department of Computer Science and Engineering, JSS Science and Technology University, Mysuru, Karnataka, India*

Abstract: ***This project simulates the complete cybersecurity attack and defense lifecycle within a controlled virtual environment using Kali Linux and Windows 10, alongside tools such as Metasploit, Sysmon, and Splunk. Virtual machines are deployed in VirtualBox on an isolated internal network, ensuring safe and realistic network interactions. The simulation begins with reconnaissance using Nmap to identify potential vulnerabilities, followed by payload creation in Metasploit to establish unauthorized access via a reverse shell. Defensive strategies include the deployment of Sysmon for detailed system activity logging and the use of Splunk for real-time monitoring and alert generation. This setup effectively replicates real-world attack scenarios and demonstrates the importance of proactive detection, incident response, and continuous monitoring. The project concludes with recommendations for enhancing organizational security postures through integrated threat detection and regular simulation exercises. By combining offensive and defensive methodologies in a hands-on lab, this work offers a practical framework for cybersecurity training and reinforces the need for adaptive defense strategies in today's evolving threat landscape.***

I. INTRODUCTION

In the contemporary digital environment, cyber threats have not only increased in frequency but have also evolved in complexity. Both organizations and individuals are consistently at risk, as cyber attackers develop increasingly sophisticated methods to exploit vulnerabilities, exfiltrate sensitive data, and disrupt standard operations [8]. The rapid advancement of technology continues to fuel this arms race, requiring cybersecurity professionals to stay abreast of both offensive and defensive strategies. The necessity for robust cybersecurity measures is now critical, and this project aims to explore and illuminate both sides of the cybersecurity equation.

This study centers on the simulation of a typical cyberattack within a controlled, isolated virtual environment. Kali Linux, which is recognized for its comprehensive suite of penetration testing tools, acts as the attack platform, while Windows 10 serves as the target [11]. Both the systems are deployed in VirtualBox, configured on an internal network to ensure realism while maintaining security and containment [2][4].

The process begins with a reconnaissance phase utilizing Nmap, a powerful tool for network exploration and vulnerability identification. Through Nmap, the Windows 10 machine is scanned to reveal open and closed ports, providing insight into potential attack vectors. Identified vulnerabilities may then be leveraged during the exploitation stage, which utilizes the Metasploit framework to craft and deliver an exploit payload. This payload, once executed on the Windows 10 virtual machine, establishes a callback to the Kali Linux machine, grating unauthorized access to the compromised system [11].

From a defensive perspective, the simulation incorporates Sysmon and Splunk for monitoring and analysis of security events [1]. Sysmon, a Windows utility, records detailed system activity—including process creation, network connections, and file modifications—while Splunk aggregates and analyses these logs to provide real-time visibility into system behaviour [12]. The combination of these tools enables defenders to detect malicious execution, trace attacker actions and monitor system alterations with considerable precision [3][5].

By simulating the full lifecycle of a cyberattack, this project emphasizes the importance of both offensive and defensive measures in cybersecurity [8]. It demonstrates the methods by which attackers exploit technical weaknesses and highlights the efficacy of monitoring and analytics tools in enabling timely detection and response [6]. Ultimately, these findings reinforce the critical role of proactive monitoring and incident response for safeguarding information systems, offering valuable insights for both cybersecurity practitioners and scholars.The project also highlighted best practices in threat detection and response, reinforcing the importance of continual learning and adaptation in the ever-changing field of cybersecurity.

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

II. PROBLEM STATEMENT

In today's digital environment, cyber threats have become alarmingly prevalent, presenting substantial risks to both organizations and individuals. Malicious actors actively seek out weaknesses in systems, exploiting them to gain unauthorized access and compromise sensitive data. Quite often, these intrusions occur without the knowledge of the intended targets, leaving defenders ill-prepared and frequently unaware until significant damage has already been done.

III. OBJECTIVE OF THE PROJECT

1) Simulating a Cyberattack

The project's primary goal is to simulate a realistic cyberattack within a fully controlled virtual environment. Kali Linux serves as the attacking system, while Windows 10 is designated as the target. The entire simulation is contained using VirtualBox's internal networking features, ensuring the exercise remains isolated from the external networks [2][4]. Kali Linux is selected for its comprehensive suite of penetration testing tools, enabling an authentic recreation of the attack lifecycle: reconnaissance, exploitation, and post-compromise activities [11].

2) NetworkScanning

The next phase involves conducting reconnaissance via Nmap. This tool enables systematic identification of open, closed, and filtered network ports on the Windows 10 machine. The results of this scan inform the subsequent steps, revealing potential vulnerabilities that may be exploited later in the exercise [11].

3) Malware Creation and Deployment

Subsequently, Metasploit is employed to generate and deploy a customized malicious payload. Typically, this may involve creating a reverse shell or similar artifact that leverages vulnerabilities identified during the scanning phase. Delivery methods mimic real-world attack vectors, such as phishing or file-sharing. Successful deployment and execution demonstrate practical control over the compromised system [11].

4) System Activity Monitoring

To monitor the effects of the simulated attack, Sysmon is installed and configured on the Windows 10 host. Sysmon logs granular system events, including process creation, file modifications, and network connections. Custom rules are configured to ensure all relevant security events related to malware execution are captured, supporting effective detection and analysis [1][12].

5) Security Event Analysis

Splunk is utilized to aggregate and visualize Sysmon logs in near real-time [3]. Through dashboards and reporting features, Splunk enables detection of malicious activity, tracking of attacker behaviour, and identification of system modifications. This analysis contributes to a comprehensive understanding of the attack's impact and supports incident response strategy development [5].

6) Improving Incident Response

The effectiveness of detection and response mechanisms is critically assessed under simulated attack conditions. This process highlights any weaknesses in existing defenses and identifies areas for improvement, such as refining detection rules or enhancing access controls. Recommendations are generated to strengthen the overall security posture, emphasizing the importance of iterative testing and continuous improvement.

7) Enhancing Learning Outcomes

Overall, the simulation provides participants with practical, hands-on experience using both offensive and defensive cybersecurity tools, including Kali Linux, Nmap, Metasploit, Sysmon, and Splunk [11]. This approach bridges the gap between theoretical concepts and real-world application, fostering a deeper understanding of cyberattack methodologies and defense strategies within a safe, controlled environment [1][12].

IV. RELATED WORKS

Several studies have explored the use of virtual environments and open-source tools for enhancing cybersecurity detection and training.

Woo-Jin Joe and Hyong-Shik Kim [1] proposed a host-based malware detection system using Windows event logs, emphasizing the importance of fine-grained system logging. Kumar and Tlhagadikgora [2] demonstrated internal network penetration testing using freely available tools such as Nmap and Metasploit, showcasing the practical application of offensive tools in controlled testbeds.

Tom Ueltschi [3], in his Botconf presentation, outlined a method for advanced incident detection and threat hunting using Sysmon logs visualized in Splunk, presenting a strong case for combining endpoint telemetry with SIEM platforms. Aoki and Suzuki [4] introduced a simplified lab environment for real-world offensive security education, primarily focusing on enabling students to understand attack methods in a safe, sandboxed setup.

Dadiyala et al. [5] proposed a structured cybersecurity home lab integrating multiple tools like Sysmon and Splunk to demonstrate detection and monitoring strategies for simulated attacks. Vielberth et al. [6], in their systematic study of Security Operations Centers (SOCs), discussed current architectures and presented open challenges, including real-time alert correlation and analyst workload optimization.

Christian Bassey et al. [7] emphasized the development of scalable SOC infrastructures using open-source tools such as OSSEC and the ELK stack, proposing a cost-effective alternative to commercial solutions. Similarly, Ueltschi's framework [3] reinforces the practical use of Sysmon and Splunk in dynamic environments for actionable detection and response capabilities.

These works collectively inform the architecture and strategy of this project, which aims to combine the strengths of these individual efforts into a unified, real-time SOC simulation that balances offensive, defensive, and analytical cybersecurity practices.

Table X: The comparison of the base paper with

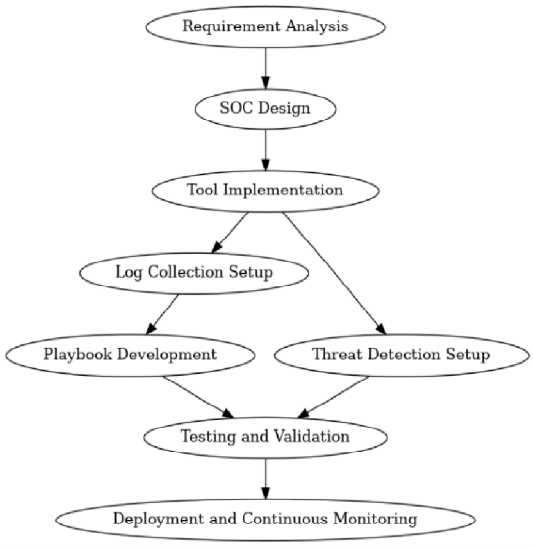
Reference / Paper	Focus Area	Detection Tools Used	Environment / Setup	Limitations
Woo-Jin Joe & Hyong-Shik Kim (2022)	Host-Based Malware detection	Windows Logs, Host Monitoring	Standalone Windows machine	No real-time SIEM integration, limited simulation control
Kumar N. S., Kgomotso Tlhagadikgora (2018)	Internal network penetration testing	Open-source scanners (Nmap, Metasploit)	Internal test network using open-source tools	Lacks post-attack detection/log analysis component

Reference / Paper	Focus Area	Detection Tools Used	Environment / Setup	Limitations
Tom Ueltschi (2025, Botconf)	Threat hunting with Sysmon and Splunk	Sysmon, Splunk	Advanced enterprise-level SOC lab	High setup complexity, targeted for experienced analysts
Yasuhiro Aoki & Katsuya Suzuki (2014)	Offensive security education lab	Custom VM setups, OSINT tools	Virtual testbed for student exercises	Focused on training, lacks integrated SIEM/log correlation
Charanjeet Dadiyala et al. (2023)	Cybersecurity home lab design and implementation	Sysmon, Splunk, Metasploit	VirtualBox-based SOC simulation	Less emphasis on AI-based or automated detection workflows
Manfred Vielberth et al. (2022, IEEE Access)	SOC architecture analysis & open challenges	Conceptual SOC Models	Theoretical framework + SOC implementation	No real or practical simulation setup provided
Christian Bassey et al. (2024)	Scalable SOC using open-source tools	OSSEC, Snort, ELK Stack	Open-source SOC stack	Focused more on scalability than hands-on attack simulation

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

V. SYSTEM DESIGN AND IMPLENTATION

A. *Flowchart*



1) Setup Environment

To begin, a controlled and isolated laboratory environment must be established. VirtualBox is employed to deploy twovirtual machines: one running Kali Linux (the attacker system), and the other running Windows 10 (the target) [11]. Both VMs are configured on an internal network within VirtualBox, ensuring communication is restricted to the lab environment and no external internet access is permitted [2][4]. This configuration mitigates unintended security risks, allowing for the safe simulation of attack scenarios.

2) Network Scanning

The next phase involves reconnaissance. Utilizing Nmap from the Kali Linux VM, a comprehensive scan is performed against the Windows 10 VM. This process reveals open, closed, and filtered ports, identifies active services, and enables OS fingerprinting. The resultant data guides the subsequent selection of potential attack vectors, closely mirroring the preliminary steps undertaken in real-world penetration testing [11].

3) Malware Creation & Exploitation

In this stage, the crafting and preparing a malicious payload using the Metasploit Framework. A suitable payload (for example, windows/meterpreter/reverse_tcp) is selected or customized, with relevant parameters such as LHOST and LPORT specified. The payload is then transferred to the target Windows VM via methods like a Python HTTP server,or simulated USB delivery. This stage emulates the tactics adversaries employ to develop and stage malware prior to execution [11].

4) Malware Deployment

The prepared payload is now deployed on the target system. On the Windows 10 VM, the attacker accesses and executes the payloadeither manually or by simulating a social engineering scenario (e.g., a user clicking a malicious file). Successful execution typically results in a reverse shell or similar exploit, closely replicating the mechanisms by which malware is introduced and activated within real-world environments.

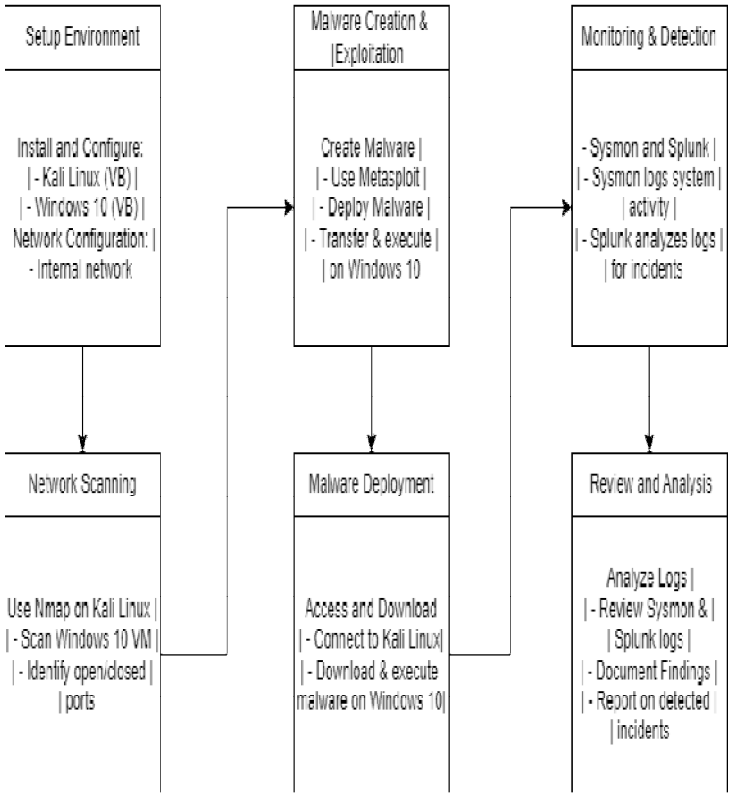
5) Monitoring & Detection

Defensive monitoring is then implemented. Sysmon (System Monitor) is configured on the Windows VM to capture detailed logs of process creation, file modifications, and network connections [1][12]. These logs are ingested by Splunk for real-time parsing and analysis, enabling the detection of anomalous or suspicious activity. This phase mirrors the functions of a Security Operations Center (SOC) utilizing SIEM tool for active threat detection [5][7].

6) Review and Analysis

Finally, the collected logs from Sysmon and Splunk are systematically analysed to reconstruct the timeline of events, confirm indicators of compromise, and identify the attack vector utilized. The analysis is documented in a formal incident report, detailing the attacker's methodology, the malware's behaviour, detection mechanisms, and lessons learned [1][3].

B. System Architecture



C. System Testing and Result Analysis

System Testing

System testing represents a pivotal stage in the software development lifecycle, focused on evaluating the integrated system to ensure alignment with specified requirements. This phase verifies the collective interaction of components, confirming that the system delivers intended functionality under practical conditions.

The current project prioritizes four central domains for system validation:

1) Network Connectivity

Establishing secure communication channels between virtual machines (VMs) is foundational within cybersecurity environments. Here, both Kali Linux and Windows 10 VMs are configured within an isolated VirtualBox network, strictly segmented from external networks to prevent unintended data exposure or external interference [11]. Network connectivity is assessed through direct communication tests—specifically, executing ping commands from Kali Linux to the Windows 10 machine. This confirms the integrity of VM isolation and internal communication, ensuring the containment of potential threats during testing[14].

Testing Summary:

- Total Connectivity Tests: 10
- Successful Connections: 9

2) Port Scanning

Port scanning is essential for assessing the security stance of target systems. Utilizing Nmap from the Kali Linux VM, the Windows 10 VM undergoes comprehensive port analysis.

The objective is to identify open, closed, and filtered ports—each revealing insights into available services and potential vulnerabilities [11]. The accuracy of these findings is verified through repeated scans, offering a reliable foundation for subsequent exploitation or defense strategies.

Testing Summary:

- Total Scans: 8
- Successful Scans: 7

3) Malware Deployment

Malware deployment tests the system's resilience against realistic attack scenarios. The project employs the Metasploit framework to craft and deliver a payload from the Kali Linux VM to the Windows 10 VM. This process simulates typical cyberattacks, such as reverse shells or remote code executions, and assesses the effectiveness of native Windows defences [11]. Key indicators—like process creation, file manipulation, and network activity—are monitored to evaluate payload execution and system response.

Testing Summary:

- Total Deployments: 6
- Successful Deployments: 5

4) Incident Detection

Effective incident detection is vital for timely threat identification and response [5]. Sysmon is deployed on the Windows 10 VM to capture granular system event data, including process launches, file changes, and network connections [5]. These logs are forwarded to Splunk for real-time analysis. The testing focus is twofold: verifying that Sysmon accurately records relevant events and ensuring Splunk's capability to ingest, index, and analyse this data. Additionally, custom Splunk rules are evaluated for their effectiveness in triggering alerts upon suspicious activities [12][3].

Testing Summary:

- Total Detection Tests: 8
- Successful Detections: 7

5) Response to Exploits

In assessing the system's response to exploits, practical testing was conducted by deploying malware within a Windows 10 virtual machine to observe the performance of Sysmon and Splunk [1]. The primary focus was on evaluating detection speed and accuracy—specifically, how efficiently the system could identify and alert on exploit activity [3][5].

Key indicators of compromise were monitored throughout, including unauthorized process initiation, unusual network activity, and unexpected modifications to system files. Special attention was paid to the time elapsed between the execution of the malware and Splunk's generation of an alert. Additionally, the system's forensic capabilities were scrutinized to ensure comprehensive documentation of relevant data, such as IP addresses, process IDs, and timestamps—crucial elements for effective incident analysis [3][5].

A robust and timely response is critical for enabling security teams to intervene before significant damage occurs. Results from the testing are summarized below:

Testing Summary:

- Total Response Conducted: 10
- Successful Responses: 9

D. Result Analysis

The results of the system testing provide valuable insights into the effectiveness of the implemented setup, the robustness of the security measures, and the accuracy of the detection mechanisms[15]. The following sections discuss the primary outcomes in detail:On the defensive front, the integration of Sysmon and Splunk proved invaluable [1]. Sysmon offered detailed event loggingincluding process creation, file modifications, and network activitywhile Splunk enabled the real-time ingestion, analysis, and alerting of these logs [3]. This combination underscored the critical importance of proactive monitoring and rapid incident response in minimizing the impact of potential threats [5].

1) Network Connectivity and Isolation

VirtualBox Environment	
Kali Linux	Windows 10
Attacker machine	Target machine

Network Configuration	
Internal network	Internal network
192.168.20.11	192.168.20.10
ping 192.168.20.10	ping 192.168.20.11

2) Port Scanning

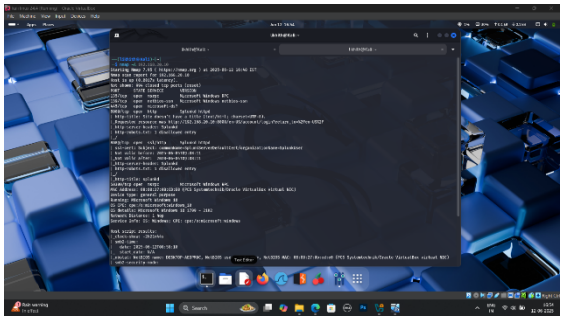


Fig 4.2.2 Open/Closed ports scanning on Win 10

3) Malware Deployment and Execution

Fig 5.4.3 Exploit the malware file.

4) Sysmon and Splunk Detection

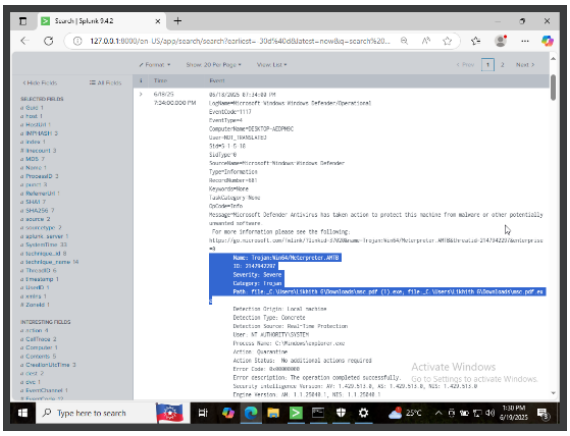


Fig 6.3.1 By analysing the security events we get to know about the malware [1][3].

5) Incident Response and Alerts

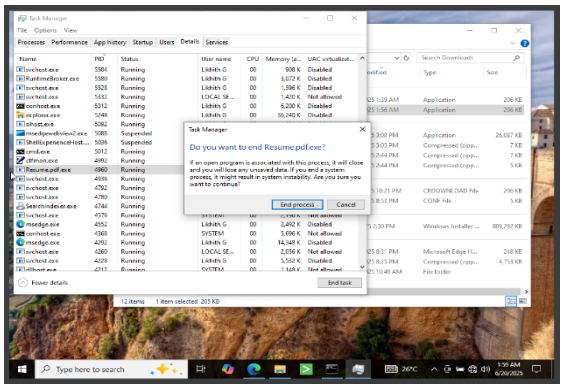


Fig 6.34 Selecting the malware file and ending the process

In summary, the testing confirmed that the system's configuration and security controls were effective across all critical stages, from network isolation to incident response. The results validate the system's suitability for controlled cybersecurity testing and best practices for similar environments[6][8].

VI. CONCLUSION

The conclusion of the SOC Home Lab project highlights the effective simulation of a realistic cybersecurity attack and defense lifecycle within a secure, virtualized environment. By leveraging VirtualBox to isolate Kali Linux (as the attacker) and Windows 10 (as the target) on an internal network, the project established a controlled platform for safe and authentic security testing.

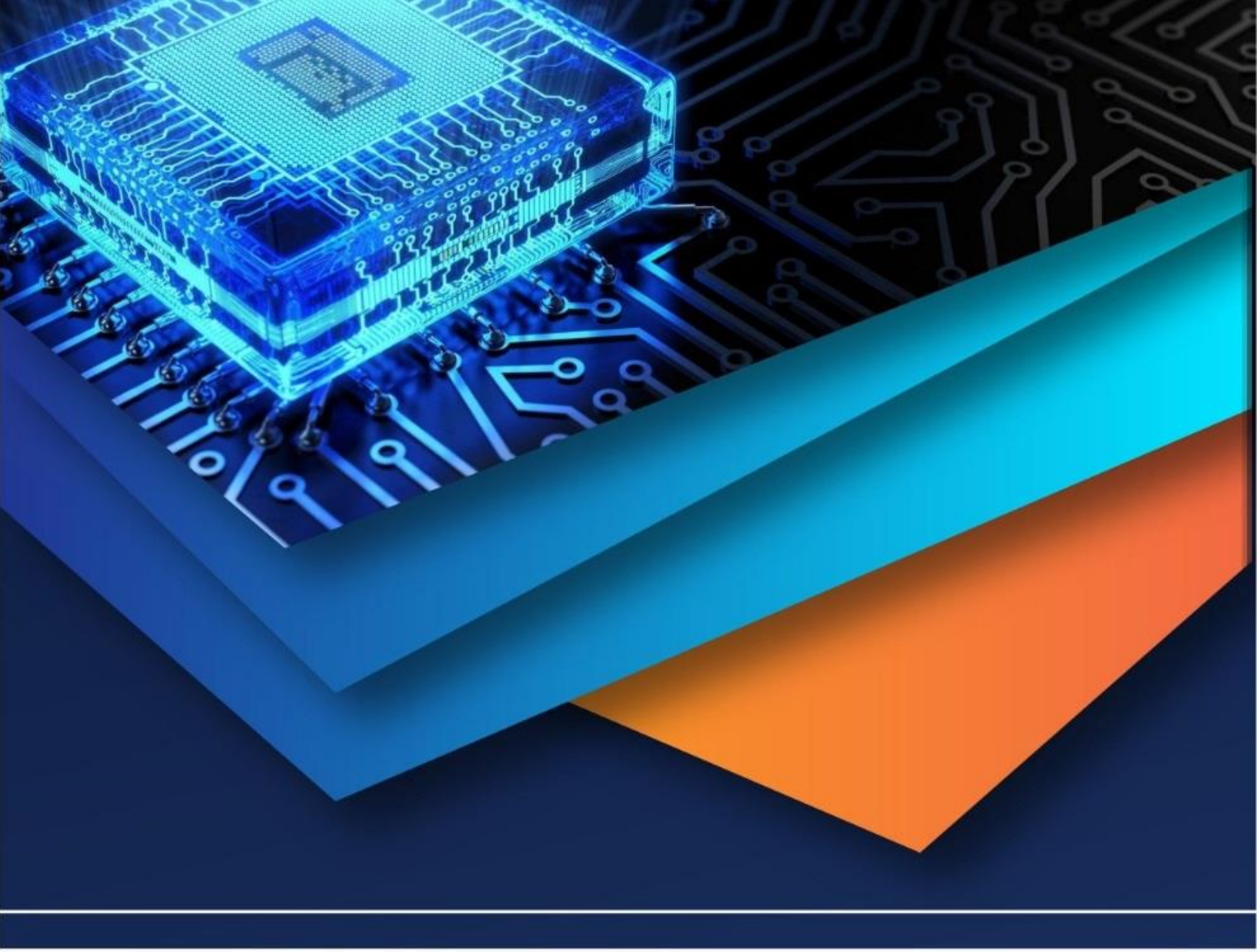
The exercise commenced with network reconnaissance using Nmap, which identified open and closed ports on the Windows 10 virtual machine, thus exposing potential vulnerabilities. Subsequently, Metasploit was utilized to craft and deploy tailored malicious payloads, successfully demonstrating how attackers can establish unauthorized access through reverse shell connections.

On the defensive front, the integration of Sysmon and Splunk proved invaluable. Sysmon offered detailed event logging—including process creation, file modifications, and network activity—while Splunk enabled the real-time ingestion, analysis, and alerting of these logs. This combination underscored the critical importance of proactive monitoring and rapid incident response in minimizing the impact of potential threats.

The project culminated in a comprehensive reconstruction of the attack timeline, identification of indicators of compromise (IOCs), and a thorough assessment of the Windows 10 system's security posture. These findings emphasize the necessity for organizations to conduct regular attack simulations and to maintain robust monitoring frameworks to strengthen their cybersecurity defenses and preparedness against evolving threats.

REFERENCES

[1] Woo-Jin Joe, Hyong-Shik Kim. "Host-Based Malware Variants Detection Method Using Logs." Journal of Information Processing Systems (JIPS), ISSN: 2092-805X, 2022.Available at: https://jips-k.org/pub-reader/632
[2] Kumar N. S., Kgomotso Tlhagadikgora. "Internal Network Penetration Testing UsingFree/Open-Source Tools: Network and System Administration Approach." In: AdvancedInformatics for Computing Research: Second International Conference, ICAICR 2018,Shimla, India, July 14–15, 2018, Revised Selected Papers, Part II, Communications inComputer and Information Science, Vol. 955, Springer, 2018. Print ISSN: 1865-0929 /Electronic ISSN: 1865-0937. https://link.springer.com/chapter/10.1007/978-3-030-12832-7_24
[3] Tom Ueltschi (Swiss Post CERT). "Advanced Incident Detection and Threat Huntingusing Sysmon and Splunk." Botconf 2025 Conference Presentation. Availableat: https://www.botconf.eu/botconf-presentation-or-article/advanced-incident-detection-and-threat-hunting-using-sysmon-and-splunk/
[4] Yasuhiro Aoki, Katsuya Suzuki. "A Simple Laboratory Environment for Real-WorldOffensive Security Education." In: Proceedings of the 2014 Information SecurityConference (ISC 2014), 2014. Availableat:https://www.researchgate.net/publication/270152499_A_Simple_Laboratory_Environment_for_Real-World_Offensive_Security_Education
[5] Charanjeet Dadiyala, Prasanna Tangade, Gaurav Singh. "Designing and Implementingan Effective Cybersecurity Home Lab for Detection and Monitoring." In: Proceedings of the14th International Conference on Computing, Communication and NetworkingTechnologies(ICCCNT 2023), 2023. (Foraccess/search:https://ieeexplore.ieee.org/Xplore/home.jsp)
[6] Manfred Vielberth, Fabian Böhm, Ines Fichtinger, Günther Pernul. "Security Operations Center: A Systematic Study and Open Challenges." IEEE Access, ISSN: 2169-3536, 2022.Available at: https://ieeexplore.ieee.org/document/98296846
[7] Christian Bassey, Ebenezer Tonye Chinda, Samson Idowu. "Building a Scalable SecurityOperations Center: Focus on Open-source Tools." Journal of Engineering Research andDept. of CS&E 50Adaptive Cyber Defense Lab: Realistic SOC Workflow implementationReports, 2024, Vol.26, Issue 7, pp.196-209. ISSN: 2582-2926 (general). Availableat: https://journaljerr.com/index.php/ JERR/article/view/1203
[8] Shahroz Tariq, Mohan Baruwal Chhetri, Surya Nepal, Cecile Paris. "Alert Fatigue inSecurity Operations Centres: Research Challenges and Opportunities." ACM ComputingSurveys, ISSN: 0097-8493, 2025. DOI: 10.1145/3723158. Availableat: https://dl.acm.org/doi/10.1145/3723158
[9] https://www.researchgate.net/publication/375880187_Designing_and_Implementing_an_Effective_Cybersecurity_Home_Lab_for_Detection_and_Monitoring
[10] https://medium.com/@efamharris/my-first-attempt-at-building-a-simple-home-lab for-threat-detection-and-monitoring-a0e6513e5432
[11] https://www.geeksforgeeks.org/using-metasploit-and-nmap-to-scan-for vulnerabilities-in-kali-linux/?
[12] https://www.researchgate.net/publication/361991727_Revisiting_the_Detection_of_Lateral_Movement_through_Sysmon
[13] https://docsdrive.com/?pdf=medwelljournals% 2Fjeasci%2F2017%2F8723 8729.pdf
[14] Prasanna B.T., Ramya, D., Shelke, N. et al. Radial basis function neural network-based algorithm unfolding for energy-aware resource allocation in wireless networks. Wireless Netw **30**, 7041–7058 (2024).
[15] Prasanna B. T, and C.B. Akki. "Dynamic Multi-Keyword Ranked Searchable Security Algorithm Using CRSA and B-Tree." Int. J. Comput. Sci. Inf. Technol, IJCSIT, Vol 6 (2015): 826-832



10.22214/IJRASET

45.98

IMPACT FACTOR: 7.129

IMPACT FACTOR: 7.429

INTERNATIONAL JOURNAL FOR RESEARCH

IN APPLIED SCIENCE & ENGINEERING TECHNOLOGY