Journal of Information & Optimization Sciences
ISSN 0252-2667 (Print), ISSN 2169-0103 (Online)
DOI : 10.47974/JIOS-1981

Innovative host-based intrusion detection systems : Safeguarding networks in the digital Era

Himjyoti Talukdar [‡]
Nitish Kumar Sarma [†]
Department of Computer Science
Kristu Jayanti College
Bengaluru
Karnataka
India

Debashish Bordoloi [§]
Department of Computer Science and Engineering
Rajiv Gandhi University
Itanagar
Arunachal Pradesh
India

Riya Sil *
Nirmalya Pal [@]
Department of Computer Science and Engineering
Brainware University
Kolkata
India

Abstract

The increasing frequency of attacks on computer systems, networks, and information systems, in general necessitate robust defense mechanisms and that can be detected by Intrusion-Detection Systems (IDS). IDS, which notify system administrators of potential intrusions in a system. As the scope and intricacy of network attacks continue to grow, there has been a clear shift in focus towards the network itself, rather than individual operating systems. Combatting these attacks through network-based intrusion detection poses its

[‡] *E-mail:* **himzyotitalukdar@gmail.com**
[†] *E-mail:* **nitishsarma8@gmail.com**
[§] *E-mail:* **debashishbordoloi007@gmail.com**
* *E-mail:* **riyasil1802@gmail.com (Corresponding Author)**
[@] *E-mail:* **nirmalya03561@gmail.com**

© Taru Publications

own set of challenges, including the management of the immense amount of data generated during network auditing. Moreover, the complex nature of these attacks often results in related events being spread out across various network locations. To overcome these hurdles, this article proposes a fresh and seamless technique for detecting intrusions. Fueled by a combination of frameworks and the Python programming language, this method aims to enhance the effectiveness of intrusion detection systems by addressing the intricacies inherent in network attacks. The system's defense against viruses and unauthorized network or system access is its main objective. Network-based IDS and host-based IDS are the two different categories of intrusion detection systems. Furthermore, the article discusses the integration of this host-based IDS method within existing security frameworks, emphasizing its adaptability and compatibility with diverse system architectures. The exploration of this novel approach provides valuable insights into the ongoing efforts to fortify information systems against evolving threats, showcasing the importance of innovation in the field of intrusion detection. In summary, this research contributes to the ongoing discourse on intrusion detection by proposing a novel method that aligns with contemporary security needs. The implementation of this method promises to reinforce the defenses of computer systems against the ever-evolving landscape of cyber threats.

Subject Classification: *68M25, 68M10, 68M12.*

Keywords: *Host-based IDS, Network security monitoring, Security alerts, Scapy, Threat detection, Anomaly, Network traffic analysis.*

1. Introduction

Due to the extensive use of computer systems, Intrusion Detection Systems (IDS) have taken computer security very seriously. The challenges of how to securely transmit and protect digital information is complicated due to the internet's rapid development and usage. Host Based Intrusion Detection System (HIDS) is the original type of IDS, and its primary function is internal monitoring (inside a computer or other equipment). However, various HIDS variants have arisen that can be used to monitor networks [1]. In the near future, intrusion detection systems will become a crucial area of research because it is hard to build a safe system without vulnerabilities. A completely secure information system may not exist due to operational or legacy constraints. In order to detect the formation of dangerous states, intrusion-detection systems must therefore, continuously monitor the use of such technologies [2]. Systems for detecting intrusions are created to detect attacks on computer networks, systems, and information systems in general. Creating and maintaining information systems that can be proven to be secure during the course of use is extremely difficult [3]. The next-generation firewall is designed to address the complexities of network attacks by integrating an IDS module. An IDS

is a module that uses behavioral analysis, signature-based detection, and malicious activity identification to find intrusions. To put it simply, an IDS is a system that sounds an alarm whenever it detects abnormal behavior in the network environment [4]. The program detects both attempted and actual abuse by authorized users as well as outside parties abusing their rights or taking advantage of security holes. Basic rules-based algorithms are frequently used by firewalls and spam filters to accept or prohibit protocols, ports, or IP addresses. These firewalls and filters have the drawback of frequently being unable to handle intricate DoS (Denial of Service) attacks and of being unable to differentiate between "good traffic" and "bad traffic". An individual computer or server is designated as the host for a HIDS, which is deployed and only keeps track of that system's activity. Firewall technology is crucial to intrusion detection system functionality. It is designed with firewall security in mind. The firewall defends the company from harmful web-based attacks. The IDS detects attempts to access a network through the firewall. The system administrator is notified, and the id is generated whenever a malicious user tries to breach the firewall security and get access to any system inside the organization.

The key contribution of the research paper is the proposal of a novel and seamless technique for detecting intrusions in network systems. This method, leveraging a combination of frameworks and the Python programming language, aims to enhance the effectiveness of intrusion detection systems (IDS) by addressing the complexities of network attacks. The paper emphasizes the integration of host-based IDS within existing security frameworks, highlighting its adaptability and compatibility with diverse system architectures. This innovative approach seeks to fortify computer systems against evolving cyber threats, contributing significantly to the field of intrusion detection by providing valuable insights and reinforcing system defenses.

In this paper, Section 2 categories the intrusion detection systems. Section 3 provides a clear idea of the motivation. In Section 4, various research works have been discussed in the literature survey section. A clear concept of the proposed work has been explained in Section 5, followed by Section 6, where the results have been demonstrated in detail. Finally, Section 7 concludes the paper along with the future scope.

2. Categories of Intrusion Detection Systems

Intrusion detection systems mostly come in two different varieties. The medium or manner of the incursion was used to categorize. Two types of IDS are most frequently found, and they are as follows:

2.1 *Network-based Intrusion Detection Systems*

Network-based attacks are now frequent and well-developed. Because of this, intrusion detection systems are now concentrating more on the network itself rather than the hosts and their operating systems. A Network-Based Intrusion Detection System (NIDS) is recognized as a system that examines network traffic to identify potential security breaches. It focuses on examining the network data packets and flows that are routed through particular network segments or hardware. An overview of a NIDS architecture is provided below:

(i) Packet Capture - A NIDS design uses sensors to collect packets from network traffic. They gather data such as protocol information, packet payloads, source and destination IP addresses, port numbers, and other pertinent information. The analysis component receives the collected packets for further processing.

(ii) Analysis Engine - The analysis engine serves as the NIDS architecture's central element. It receives the packets that were collected and uses a variety of analytic techniques to look for any potential intrusions. To find potential threats, the analysis engine compares the collected packets to known attack signatures, patterns, or baseline behavior. NIDS commonly identifies anomalies in behavior by analyzing the flow of network traffic. Numerous network devices, including switches, routers, and even certain networked hosts, are utilized for this purpose [5].

(iii) Signature Database - A database of predetermined signatures or known attack patterns is kept up to date by the NIDS. To stay up with the changing threat landscape, these signatures are frequently updated. In order to find any matches, the analysis engine compares the collected packets to the signatures in the database.

(iv) Alert Generation - The analytical engine produces alerts or notifications when it finds questionable or malicious activity. These notifications warn of potential security lapses or unusual behavior. The alerts typically provide details about the danger that

was discovered, including the source and destination IP addresses, the type of attack, the level of severity, and any further contextual information.

(v) Reporting and Logging - Mechanisms for producing reports and keeping logs of network activities and observed events are part of the NIDS architecture. These reports and logs offer a record of past network security occurrences, help with forensic analysis, and support compliance standards.

2.2 *Host-based Intrusion Detection Systems (HIDS)*

The Host-Based Intrusion Detection System (HIDS) keeps track of the events and activities taking place on each unique host system. Only the host system on which it is installed is protected by HIDS, and its network card runs in non-promiscuous mode. In order to identify potential security breaches, it functions at the host level, examining system logs, file integrity, and other host-specific data. These IDS systems have historically relied heavily on and used the signatures of known attacks to find inbound threats.

(i) Sensor/Agent - A software element that is installed on the host system is the sensor or agent. It gathers information on network connections, file integrity, system events, logs, and other pertinent topics. Monitoring and analyzing host-level actions to find potential security threats is the responsibility of the sensor/agent.

(ii) Event Collector - Data produced by the sensor or agent is delivered to the event collector. The data is collected, normalized, and stored for later study. Before providing the data to the analysis engine, the event collector may use pre-processing techniques to filter or enrich the data.

(iii) Analysis Engine - The HIDS architecture's core element is the analysis engine. To find potential security events or malicious actions, it uses a variety of analysis approaches, including correlation, anomaly detection, behavior analysis, and signature-based detection. To find irregularities or signs of compromise, the analysis engine compares the collected data with well-known attack patterns, behavioral profiles, or regulations.

(iv) Rule Engine - A set of established rules or regulations that specify the conduct or traits of legitimate and malicious actions are contained in

the rule engine. The analyzed data is subjected to these standards in order to spot any suspicious or malicious trends. The rule engine can be modified or updated to meet certain security needs.

(v) Alerting and Notification - An alert is sent when the analysis engine or rule engine finds a security issue or suspicious conduct. Sending immediate alerts to security managers, system administrators, or a Security Information and Event Management (SIEM) system is the responsibility of the alerting and notification component. These warnings include information that can be used to start incident response or mitigation procedures.

2.3 *Scope of Host-based Intrusion Detection System*

The main objective of a HIDS for network monitoring is to monitor and safeguard specific hosts or endpoints within a network environment. HIDS and Host Intrusion Prevention System (HIPS) are implemented at endpoints, in contrast to NIDS, which are employed at network boundaries and segmentation points like gateways to the internet or other untrusted networks [6] As opposed to host-based IDS systems, which function at the host level and offer granular visibility and security for particular devices, network-based IDS systems primarily analyze network traffic. The intrusion detection paradigm that is used may be used to further categorize HIDS. The misuse detection technique and the anomaly (or behavior) detection approach are two examples of these categories [7]. HIDS recognizes unknown attacks more often than network-based IDS because the attack disrupts the host being monitored. Suspicious actions coming from within the system, such as unauthorized access attempts, privilege escalation, or unusual behavior by privileged users, can be found using HIDS. Before serious damage is done, it can detect unwanted software, such as malware or ransomware, and notify administrators. The following are the main characteristics of HIDS in terms of network monitoring:

(i) Endpoint Protection - Individual endpoints, including servers, workstations, laptops, and other network-connected devices, are the main focus of HIDS monitoring and protection. It seeks to identify and stop security problems, malware infections, unauthorized access, and intrusions that target these particular hosts.

(ii) Threat Detection - To identify risks and security incidents, HIDS makes use of a variety of methodologies. It uses heuristic algorithms, behavior analysis, anomaly detection, signature-based detection,

and anomaly detection to spot potential signs of compromise as well as recognized attack patterns. When suspicious behavior or malicious events are found, it issues alerts or initiates reactions.

(iii) Incident Response Support - When a security issue occurs, HIDS plays a crucial role in incident response by providing real-time alerts and useful information. It helps security teams locate the issue's origin, comprehend the scope of the compromise, and enable efficient incident containment, eradication, and recovery.

3. Motivation

Although several efforts have been made in the area of security, there is still much effort required for improvement. No system has been constructed with 100% security; all attacks and attempted intrusions have contributed some security weaknesses to every system. Every day, several systems are attacked by numerous viruses and other forms of malware.

Government, finance, and the healthcare sectors all have very strict regulatory compliance standards. Host-based IDS can assist in meeting these objectives by offering capabilities for continuous monitoring and auditing. It enables organizations to prove compliance by quickly identifying and resolving security incidents. In the realm of small businesses and individuals with limited financial and technical resources, there is a pertinent need for effective yet affordable security solutions. Implementing a less sophisticated but still valuable Intrusion Detection System (IDS) can offer crucial insights. Open-source IDS, for example, provides a cost-effective option for monitoring network traffic, detecting suspicious activities, and identifying known attack patterns. These capabilities can be deployed on a dedicated server or a network gateway, providing a layer of defense for those with constrained resources. The specific focus of this study is to introduce a technique for identifying open ports and scrutinizing them for potential malicious network activity. This approach aims to contribute to the development of a low-cost, yet robust, host-based intrusion detection system. By concentrating on monitoring networks on a specific host where the system is installed, this strategy seeks to enhance the overall security posture of systems, especially those with limited resources and capabilities. The research aspires to provide a practical solution to the ongoing challenge of fortifying computer networks against evolving cyber threats. Implementing a less sophisticated but still useful intrusion detection system (IDS) can give small firms with limited financial and technical resources important security insights. An

inexpensive option might be an open-source IDS. Small businesses and individuals can monitor network traffic for suspicious activity and known attack patterns with the use of basic network-based intrusion detection capabilities, which can be installed on a dedicated server or a network gateway. This study aims to present a technique for identifying open ports and checking them for any malicious network activity. It could be useful in creating a low-cost, simpler host-based intrusion detection system to keep an eye on networks on a specific host where the system is installed.

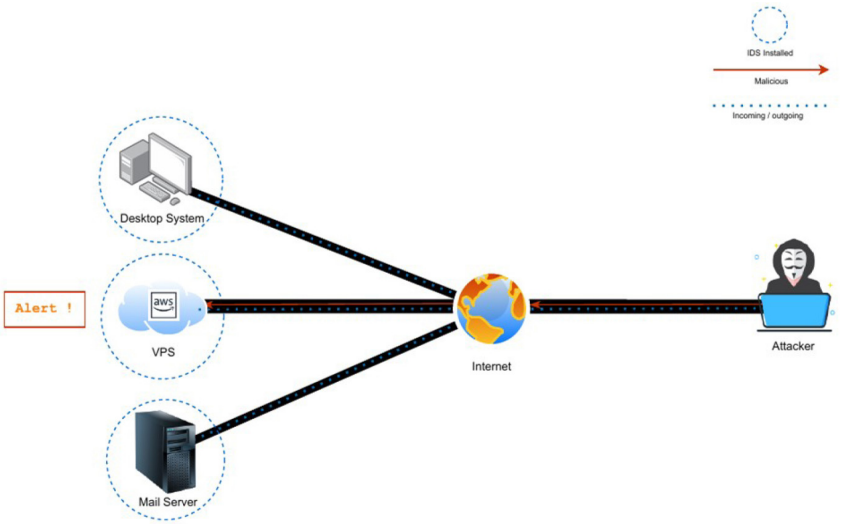


Figure 1

Depicts the authors' designed IDS in action

Figure 1 depicts the authors' designed IDS in action. The system is under a simulated attack, where an adversary is attempting to compromise the system. The IDS, embedded within the system, successfully identifies the malicious activity and triggers an alert, showcasing the system's capability to detect and respond to potential security threats.

4. Literature Survey

The difficulties of identifying unanticipated network security breaches are addressed by the deep learning-based Network Intrusion Detection System (NIDS) that uses Self-taught Learning (STL) on the NSL-KDD dataset. In terms of accuracy, precision, recall, and f-measure values,

this strategy outperforms earlier ones in terms of flexibility and efficiency [8]. The proliferation of Information Technology (IT) into our daily lives has made it possible to communicate with a wide range of data formats at several OSI model layers, ranging from application to physical. There is a significant need to protect the confidentiality, availability, and integrity of this data because they are vulnerable to intrusion hazards. A device or software program called an Intrusion Detection and Prevention System (IDPS) is made to monitor a system or network [9]. To ensure that only authorized recipients may decrypt communications while maintaining message validity, people use cryptography with secret keys for secure communication. On the other hand, hackers might use advanced instruments to target network hosts and obtain private information. IDS protect users in these kinds of situations by performing vital security tasks including monitoring, identifying, and reacting to illegal activity [10]. The literature on host-based intrusion detection systems (HIDS) is organized according to the data sources; system logs, audit data, Windows Registry, file systems, and program analysis are utilized to give targeted surveys of HIDS research. Even though system calls are usually included in audit data, IDS research has been sparked by the availability of publicly available system call datasets, therefore it deserves special attention [11]. Intrusion detection systems (IDS) are becoming essential for information assurance due to the increase in cyber-attacks. IDS keep an eye on and collects data from a secured system, evaluates and correlates the data, and initiates actions when it finds evidence of an intrusion [12]. Since recorded snapshots can be separately evaluated, it would be interesting to investigate the scalability of the analysis engine. Current intrusion detection systems have several constraints. Their main methods for analyzing network traffic include statistically anomalous traffic patterns or scanning for known attack patterns. But because traffic encryption technologies like SSL and IPSEC are so widely used, a large amount of internet data is encrypted and cannot be examined. There are many well-known methods of evading network intrusion detection systems. As a result, evasion strategies are unsuccessful and intrusion detection is rapidly moving to host servers where content is visible [13]. The study addresses the use of statistical data evaluation to examine anomalies in user behavior; however, it does not include particular method specifics, which makes it difficult to evaluate the resilience and reliability of the intrusion detection system. The accelerated single testing step, intended to improve performance, presents questions regarding possible overuse of system resources, which could have a detrimental effect on user experience. There is still research to be

done on the trade-off between accuracy and resource efficiency [14]. Modern data centers produce large amounts of real-time system call traces from Linux programs that are inappropriate for use with host-based intrusion detection solutions. It takes time to train data mining models with system calls on a single host with static processing and storage capability, and handling intermediate datasets effectively is difficult. Most systems employ signatures to identify and address usage trends, even with the development of intrusion detection techniques. However, human interaction is still required to oversee and guarantee system integrity because automated replies carry a risk [15]. Network flow metrics and biometric data together improve IDS performance in the healthcare industry, although scalability and adaptation issues could occur. Behavior-based detection strategies are necessary since traditional signature-based detection techniques have trouble identifying unknown incursions [16]. Particularly in anomaly detection, where imbalances and dynamic domains present difficulties, current intrusion detection technologies fall short of designing systems that satisfy all real-world requirements. Strong intrusion detection systems are essential, as seen by the rise in cybercrime. Although signature-based detection techniques outperform known threats, they necessitate the manual update of signature databases. When it comes to assaults with unknown signatures, they are useless. Relying too much on time-based detection could result in false negatives or positives. It is yet unknown how well IDS models operate in real-world networks and how well they can adjust to changing security threats [17]. Certain network intrusion detection and prevention settings could miss non-network attack vectors, thus ongoing monitoring and upgrades are needed to keep them working. Although real-time notifications are provided, their impact may be limited by delayed reaction times, necessitating optimization [18]. Even with advancements in attack detection, combined NIDS and HIDS methods might not be able to handle complex attacks that are not well represented by conventional models. Validation across a range of circumstances is necessary to ensure the efficacy of suggested HIDS models [19]. The difficulty of maintaining dependability in hostile environments is highlighted by the vulnerabilities that host-based intrusion detection systems have if the underlying operating system is corrupted. The independence of HIDS from the supervised system to reduce attack vectors is directly related to its resilience [20]. The security community is aware of over a billion dangerous executable scripts, according to data from the German AV-TEST institute. Actually, as a result of the digital transition, there has been a rise in

cybercrime, which is frequently linked to large financial losses for both individuals and businesses [21]. The detection of known assaults is the basis for signature-based detection techniques, also known as misuse-based detection techniques. They typically outperform attacks that are already detected by their signatures [22]. The database containing the signatures has to be manually updated with fresh signatures to guarantee they continue to function properly with few to no false alarms. As a result, they lack the ability to defend against assaults whose signatures are missing from the database [23]. Because some lawful network activity may exhibit quick connection rates, relying solely on time-based detection may lead to false positives or negatives. The IDS model's scalability and adaptability to changing threat landscapes are still ignored, and its efficacy in a real production network has not yet been demonstrated [24]. S2ipt expressly sets up ip tables to use Snort rules for network intrusion detection and prevention; it may, however, ignore non-network attack vectors like host- or file-based exploits. Maintaining current s2ipt configurations may become difficult as Snort rules change, necessitating ongoing monitoring and modifications to guarantee the system's efficacy against new threats [25]. Even though Keris offers real-time alerts, its 0.72- and 0.44-second delays for Telegram Web hook and FCM, respectively, could be a hindrance in situations where time is of the essence. As a result, additional tuning is required to achieve faster reaction times [26]. The combined NIDS and HIDS strategy may still have limits in identifying sophisticated attack types, especially those using unique and stealthy approaches that may not be sufficiently captured by conventional machine learning models, even though attack detection performance has improved [27]. The features of the recently introduced PLAID dataset may have an impact on the efficacy of the suggested HIDS models, especially ALAD, and its generalizability to a variety of real-world settings requires validation [28]. An attacker that gains control of the underlying operating system can compromise host-based intrusion detection systems. It is possible to turn off HIDS, manipulate the analysis engine, update or remove important files, modify input data, or stop producing alerts. It's difficult to guarantee a dependable input and output in a possibly hostile environment [29]. The likelihood that HIDS will be compromised in the case of an assault on the system under supervision is how Molina and Cukier define HIDS resiliency [30]. They contend that independence and resilience are intimately related [31]. Because shared system components might act as attack vectors, HIDS should be as independent of the supervised system as possible [32].

5. Proposed works

Although Scapy is essentially a packet manipulation library, it may be used to create an Intrusion Detection System (IDS) that will undoubtedly monitor and detect network intrusions and reach full IDS functions. However, implementing a whole IDS only using Scapy can be quite tough. However, using Scapy for certain tasks like packet capture, packet analysis, and rule matching will aid in creating an intrusion detection system that is completely functional. While Scapy primarily serves as a powerful packet manipulation library, its capabilities extend beyond conventional usage, making it a valuable tool for constructing a comprehensive IDS. Leveraging Scapy for the creation of an IDS introduces a unique approach to monitoring and detecting network intrusions, allowing for the realization of full IDS functionalities. Building an entire Intrusion Detection System solely with Scapy, however, presents a challenging endeavor. The complexity of implementing a robust IDS requires considerations beyond Scapy's capabilities. Despite this, strategically employing Scapy for specific tasks such as packet capture, packet analysis, and rule matching can significantly contribute to the development of a functional intrusion detection system. Incorporating Scapy into the IDS framework enhances its packet-level capabilities, enabling precise analysis and identification of potential threats. The utilization of Scapy's packet manipulation features empowers the IDS to not only monitor network traffic but also to perform in-depth analysis, aiding in the formulation of effective rules for intrusion detection. While Scapy may not serve as the sole foundation for a complete IDS, it undoubtedly serves as a valuable component, elevating the overall efficacy of the intrusion detection mechanism. An overview of how Scapy can be used as a component of an intrusion detection system is given below.

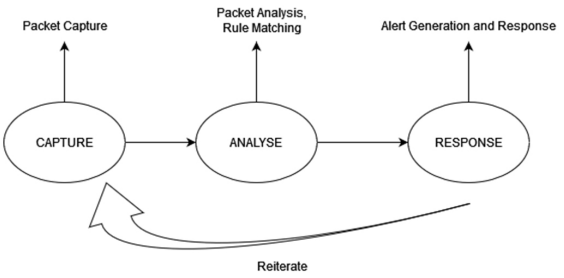


Figure 2

Overview of the core system

Figure 2 represents the core system's cyclical process for detection. It begins with the "CAPTURE" phase, where packet capture occurs to gather data from network traffic. Following this, the data moves to the "ANALYSE" phase, where packets undergo analysis, rule matching against predefined security policies, and potential anomaly detection to identify any suspicious or malicious activity. Finally, the process moves to the "response" phase, where alerts are generated. This entire cycle then repeats continuously to ensure ongoing monitoring.

(i) Detecting brute force in SSH - Utilizing Scapy, Authors can examine network data for patterns that point to SSH brute force attempts. Here's an illustration of how Scapy can be used to monitor network traffic in order to find SSH brute force assaults. For this, sniffing was implemented first, followed by packet analysis, and finally the detecting function.

(ii) Packet Capture - Scapy's packet sniffing features can be used to record network communication. Capture packets for detection -if packet. Layer (TCP) and packet. has a layer (Raw).

(iii) Packet Analysis **-** Scapy can be used to analyze captured packets and extract pertinent data, including source and destination IP addresses, ports, protocols, and payload data analyze packet payloads and headers for potential threats or unusual patterns.

Check for port 22 or SSH if tcp.dport == 22 or tcp.sport == 22: payload = str(raw.load)

(iv) Rule Matching - By comparing packet properties to specified behavior or rules that denote harmful behavior, implement rule-based pattern matching logic using Scapy. Establish a set of guidelines or signatures that describe well-known attack strategies and set off alarms or other reactions when a match is made.

Check for SSH Authentication Failure

if "SSH-2.0-" in payload and "failure" in payload:

src_ip = packet[IP].src

Count the number of logins attempt for each source IP define a threshold. If a threshold is reached, it can be considered a brute force attempt.

(v) Alert Generation and Response **-** Alerts are prioritized based on the intensity and persistence of the brute force attempts or other attacks, enabling administrators to focus their attention on the most critical

incidents first. This hierarchical approach facilitates a swift and targeted response strategy. Create alerts or take the necessary steps when suspicious activity is found. Sending alerts, keeping track of events, stifling traffic, or initiating incident response protocols can all be part of this.

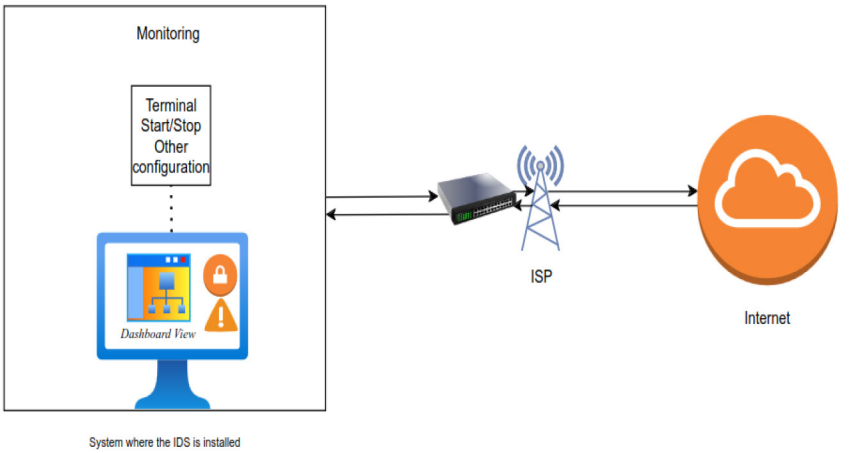


Figure 3

IDS system from surface level view

In the depicted diagram in figure 3, a surface-level view shows that the Intrusion Detection System (IDS) is situated on the far left, overseeing network traffic, accompanied by a dashboard view below for system management, centrally the Internet Service Provider (ISP) orchestrates connection to the broader internet network, and on the far right the internet links to external servers and services, enabling communication and data exchange.

6. Results

For demonstration reasons, Authors developed a simple dashboard console program to track, examine and identify brute force in SSH. This application will display open ports, alert ports and the originating IP. The objective of this study was to develop an IDS utilizing Python to monitor and detect brute force attacks on SSH connections. In order to evaluate the effectiveness of the IDS, a Virtual Private Server (VPS) was employed as a testing environment. The IDS successfully monitored TCP requests

associated with SSH connections, specifically focusing on identifying patterns indicative of brute force attacks.



Figure 4

Terminal view after running the system with root privileges

Figure 4 explains the process of how after installing the Python Flask environment and necessary networking and server libraries, the authors have initiated the IDS via terminal with admin privileges to run on localhost.

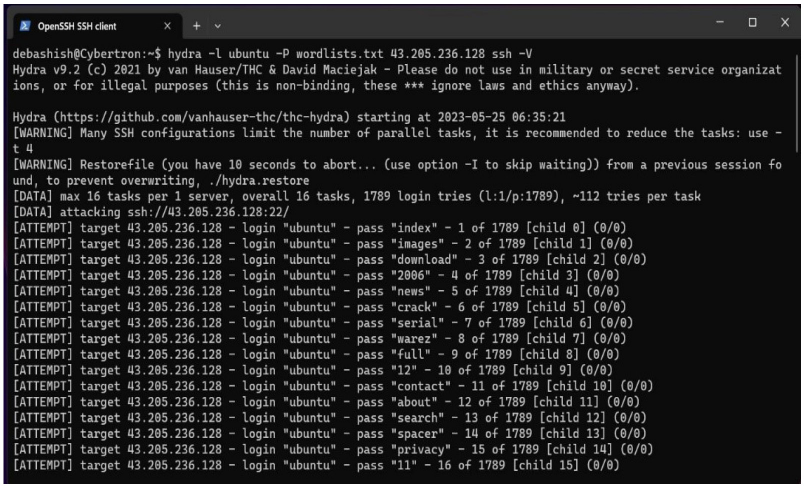


Figure 5

Testing Brute Force attack with Hydra on port22

For testing the authors have conducted a brute force attack on port 22 shown in figure 5, which is SSH from another system as an Attacker System using the tool hydra, that is used to brute force a service that is running on a system.

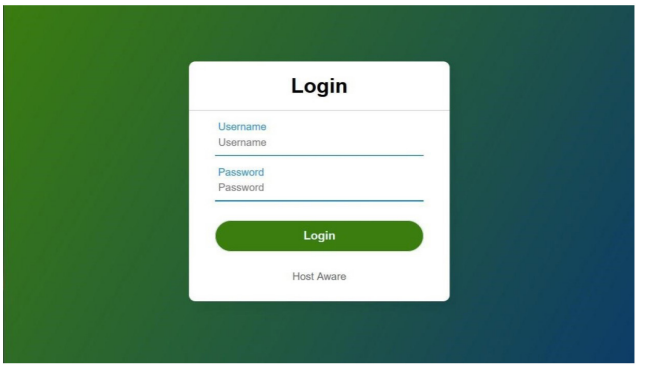


Figure 6

Login page of the proposed work

In figure 6 after running the application and navigating to the URL, the user and password page will appear, and after appearing the username and password, the page will be redirected to the dashboard page.

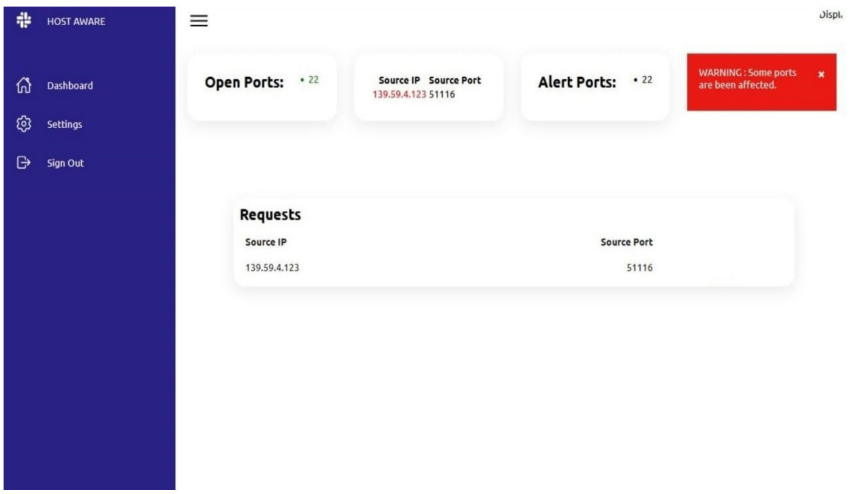


Figure 7

Dashboard view

Figure 7 consists of Open Ports, Source Ip, Source Port and alert ports on the dashboard. The alert box on the upper right corner has notified that port 22 might have been attacked. Hence authors are able to get alerts for ports that may get attacked or poked. The developed IDS successfully monitored TCP requests for SSH connection brute force attempts on a VPS, demonstrating promising results.

7. Conclusion

The creation and use of a host-based Intrusion Detection System (IDS) will significantly improve the security of computer systems, Authors have examined the many facets of host-based IDS in this study, including its foundations, methods, applications, and advantages. Through our comprehensive exploration, it is evident that the implementation of a host-based IDS offers a multifaceted approach to safeguarding computer systems. The proposed work underscores the importance of a tailored approach to security, and the potential to develop a customized host-based IDS system emerges as a proactive measure for organizations seeking an efficient security solution. The flexibility inherent in host-based IDS systems allows for the adaptation of security measures to the specific needs and vulnerabilities of a given environment. This adaptability is crucial in addressing the evolving landscape of cyber threats, where a one-size-fits-all solution may fall short.

In addition to the logic highlighted in our proposed work, the incorporation of advanced technologies, such as machine learning algorithms, could further enhance the capabilities of host-based IDS. Machine learning models can adapt to emerging threats, identify anomalies with greater accuracy, and reduce false positives, thereby refining the overall efficacy of the IDS. And also, as the Internet of Things (IoT) ecosystem expands, there is a growing need for security solutions. Host-based IDS can play a crucial role in IoT security by monitoring and detecting unusual activities within IoT devices and networks. In conclusion, the implementation of a host-based IDS not only fortifies the security posture of computer systems but also positions organizations to stay ahead in the ongoing battle against cyber threats. The continuous evolution and refinement of host-based IDS technologies will undoubtedly play a pivotal role in ensuring the resilience and adaptability of security measures in an ever-changing digital landscape.

Abbreviations

IDS	: Intrusion-Detection Systems
HIDS	: Host-Based Intrusion Detection System
DoS	: Denial of Service
NIDS	: Network-Based Intrusion Detection System
SIEM	: Security Information and Event Management
URL	: Uniform Resource Locator
STL	: Self-taught Learning
HIPS	: Host Intrusion Prevention System
SSH	: Secure Shell
IT	: Information Technology
ISP	: Internet Service Provider
VPS	: Virtual Private Server
TCP	: Transmission Control Protocol
IP	: Internet Protocol
IoT	: Internet of Things
IPSEC	: Internet Protocol Security
SSL	: Secure Sockets Layer
IDPS	: Intrusion Detection and Prevention System

Conflict of Interest Statement

The Authors declare no Conflicts of Interest.

Declarations

The authors declare that this manuscript has not been submitted to other journals for consideration.

Availability of data and materials

Not Applicable

Funding

Not Applicable

Authors' contributions

Himjyoti Talukdar: Conceptualization, Writing - Original Draft
Nitish Kumar Sarma: Conceptualization, Methodology, Coding
Debashish Bordoloi: Software, Validation
Riya Sil: Conceptualization, Methodology, Review & Editing, Supervision
Nirmalya Pal: Conceptualization, Editing

References

[1] A. Pal Singh and M. Deep Singh, "Analysis of host-based and network-based intrusion detection system," *International Journal of Computer Network and Information Security*, vol. 6, no. 8, pp. 41–47 (Aug. 2014), doi: 10.5815/ijcnis.2014.08.06.

[2] S. Kumar, S. Gupta, and S. Arora, "Research trends in network-based intrusion detection systems: A review," *IEEE Access*, vol. 9, pp. 157761–157779 (2021), doi: 10.1109/ACCESS.2021.3129775.

[3] B. Ghosh and R. Sil, "Analysis of cyber risk in finance: The state-of-the-art and future research trends," in *Contemporary Social Physics*, A. Rai, S. Karmakar, S. Chatterjee, and J. K. Pandey, Eds. Cham, Switzerland: Springer (2024), doi: 10.1007/978-3-031-77596-3_17.

[4] A. Thakkar and R. Lohiya, "A survey on intrusion detection system: Feature selection, model, performance measures, application perspective, challenges, and future research directions," *Artificial Intelligence Review*, vol. 55, no. 1, pp. 453–563 (2021), doi: 10.1007/s10462-021-10037-9.

[5] J. Liu, K. Xiao, L. Luo, Y. Li, and L. Chen, "An intrusion detection system integrating network-level intrusion detection and host-level intrusion detection," in *Proc. IEEE 20th Int. Conf. Softw. Quality, Rel. Security (QRS)* (2020), doi: 10.1109/QRS51102.2020.00028.

[6] A. Biswas, R. Sil, and A. Roy, "A study on application of interplanetary file system," in *Communication and Intelligent Systems*, H. Sharma, M. K. Gupta, G. S. Tomar, and W. Lipo, Eds., vol. 204. Singapore: Springer (2021), doi: 10.1007/978-981-16-1089-9_79.

[7] Y. Shin and K. Kim, "Comparison of anomaly detection accuracy of host-based intrusion detection systems based on different machine learning algorithms," *International Journal of Advanced Computer*

Science and Applications, vol. 11, no. 2 (2020), doi: 10.14569/IJACSA.2020.0110233.

[8] A. Javaid, Q. Niyaz, W. Sun, and M. Alam, "A deep learning approach for network intrusion detection system," in *Proc. 9th EAI Int. Conf. Bio-Inspired Inf. Commun. Technol. (BIONETICS)* (2016), doi: 10.4108/eai.3-12-2015.2262516.

[9] N. A. Azeez, T. M. Bada, S. Misra, A. Adewumi, C. Van der Vyver, and R. Ahuja, "Intrusion detection and prevention systems: An updated review," in *Data Management, Analytics and Innovation*, pp. 685–696 (2019), doi: 10.1007/978-981-32-9949-8_48.

[10] Q. Kang, L. Shi, M. Zhou, X. Wang, Q. Wu, and Z. Wei, "A distance-based weighted undersampling scheme for support vector machines and its application to imbalanced classification," *IEEE Transactions on Neural Networks and Learning Systems*, vol. 29, no. 9, pp. 4152–4165 (2018), doi: 10.1109/TNNLS.2017.2755595.

[11] R. A. Bridges, T. R. Glass-Vanderlan, M. D. Iannacone, M. S. Vincent, and Q. Chen, "A survey of intrusion detection systems leveraging host data," *ACM Computing Surveys*, vol. 52, no. 6, pp. 1–35 (2019), doi: 10.1145/3344382.

[12] Y. Ou, Y. Lin, Y. Zhang, and Y. Ou, "The design and implementation of host-based intrusion detection system," in *Proc. 3rd Int. Symp. Intell. Inf. Technol. Security Informatics (IITSI)* (2010), doi: 10.1109/IITSI.2010.127.

[13] R. Gassais, N. Ezzati-Jivan, J. M. Fernandez, D. Aloise, and M. R. Dagenais, "Multi-level host-based intrusion detection system for internet of things," *Journal of Cloud Computing*, vol. 9, no. 1 (2020), doi: 10.1186/s13677-020-00206-6.

[14] S. N. Chari and P.-C. Cheng, "Bluebox," *ACM Transactions on Information and System Security*, vol. 6, no. 2, pp. 173–200 (May 2003), doi: 10.1145/762476.762477.

[15] L. Vokorokos and A. Balaz, "Host-based intrusion detection system," in *Proc. IEEE 14th Int. Conf. Intell. Eng. Syst. (INES)* (2010), doi: 10.1109/INES.2010.5483815.

[16] M. Liu, Z. Xue, X. Xu, C. Zhong, and J. Chen, "Host-based intrusion detection system with system calls," *ACM Computing Surveys*, vol. 51, no. 5, pp. 1–36 (2018), doi: 10.1145/3214304.

[17] S. Saif, A. A. Ansari, S. Biswas, and D. Giri, "A comprehensive analysis of machine learning-based intrusion detection systems: Evaluating datasets and algorithms for internet of things," *Journal of Cyber Security Technology*, vol. 10, no. 1, pp. 1–27 (2026), doi: 10.1080/23742917.2024.2447124.

[18] M. Wolf, K. Bergner, P. Förtsch, and D. Landes, "Implications of applying emulation data to machine learning-based intrusion detection systems," *Journal of Information Security and Applications*, vol. 97, Art. no. 104362 (Mar. 2026), doi: 10.1016/j.jisa.2025.104362.

[19] P. M. S. Makhdoomi, M. Ikhlas, A. Khursheed, F. Hilal, Z. A. Najar, J. Hameed, and S. Sharma, "Network-based intrusion detection: A comparative analysis of machine learning approaches for improved security," *Journal of Cyber Security Technology*, vol. 10, no. 1, pp. 1–28 (2026), doi: 10.1080/23742917.2024.2447119.

[20] H. Bostani and M. Sheikhan, "Hybrid of anomaly-based and specification-based IDS for internet of things using unsupervised OPF based on MapReduce approach," *Computer Communications*, vol. 98, pp. 52–71 (2017), doi: 10.1016/j.comcom.2016.12.001.

[21] I. Martins, J. S. Resende, P. R. Sousa, S. Silva, L. Antunes, and J. Gama, "Host-based IDS: A review and open issues of an anomaly detection system in IoT," *Future Generation Computer Systems*, vol. 133, pp. 95–113 (2022), doi: 10.1016/j.future.2022.03.001.

[22] F. S. Masoodi, "Intrusion detection in the age of deep learning: An introduction," in *Deep Learning for Intrusion Detection*, F. S. Masoodi and A. Bamhdi, Eds. (2026), doi: 10.1002/9781394285198.ch1.

[23] O. Ajayi, A. Gangopadhyay, R. F. Erbacher, and C. Bursat, "Developing cross-domain host-based intrusion detection," *Electronics*, vol. 11, no. 21, Art. no. 3631 (2022), doi: 10.3390/electronics11213631.

[24] F. Alenezi, S. Almowuena, A. Alenezi, and M. J. F. Alenazi, "CIDS: A collaborative intrusion detection system approach for SDN-based distributed industrial plants," *Trans. Emerg. Telecommun. Technol.*, vol. 37, no. 1, Art. no. e70327 (2026), doi: 10.1002/ett.70327.

[25] X. Zhao, M. He, and X. Wang, "A real-time channel-level intrusion detection system based on multimodal learning," *IEEE Internet of Things Journal*, vol. 13, no. 7, pp. 12598–12612 (Apr. 1, 2026), doi: 10.1109/JIOT.2026.3651905.

[26] A. Tedyyana and O. Ghazali, "Real-time hypertext transfer protocol intrusion detection system on web server using Firebase cloud messaging," in *Proc. 4th Int. Conf. Appl. Sci. Technol. Eng. Sci.* (2021), doi: 10.5220/0010946300003260.

[27] L. Liu, M. Simsek, B. Kantarci, M. Bagheri, and P. Djukic, "Collaborative feature maps of networks and hosts for AI-driven intrusion detection," in *IEEE Global Communications Conference 2022*, pp. 2662–2667 (2022), doi: 10.1109/GLOBECOM48099.2022.10000985.

[28] J. H. Ring, C. M. Van Oort, S. Durst, V. White, J. P. Near, and C. Skalka, "Methods for host-based intrusion detection with deep learning," *Digital Threats: Research and Practice*, vol. 2, no. 4, pp. 1–29 (2021), doi: 10.1145/3461462.

[29] V. Bukac, P. Tucek, and M. Deutsch, "Advances and challenges in standalone host-based intrusion detection systems," in *Trust, Privacy and Security in Digital Business*, pp. 105–117 (2012), doi: 10.1007/978-3-642-32287-7_9.

[30] T. B. Ogunseyi, G. Thiyagarajan, H. He, V. Bist, and Z. Du, "Performance analysis of explainable deep learning-based intrusion detection systems for IoT networks: A systematic review," *Sensors*, vol. 26, no. 2, Art. no. 363 (2026), doi: 10.3390/s26020363.

[31] D. Talapula, A. Kumar, K. Ravulakollu, and M. Kumar, "A novel intrusion detection system in cloud infrastructure using deep learning technique," *J. Discrete Math. Sci. Cryptogr.*, vol. 26, no. 5, pp. 1277–1303 (2023), doi: 10.47974/JDMSC-1739.

[32] R. Kowtharapu, C. H. S. Kiran, and P. A. Kumari, "Dynamic intrusion detection for internet of drones using glowworm swarm optimization," *Scientific Reports* (2026), doi: 10.1038/s41598-026-44789-7.

Received October, 2024