Securing Networks with Precision: Unveiling the Potential of Application Protocol Based Intrusion Detection Systems

Shivani Karthikeyan, Shrish KS, Arunkumar J, Bagavathi C

Department of CSE, Amrita School of Computing, Coimbatore, Amrita Vishwa Vidyapeetham, India, University of India

Corresponding author: Bagavathi C, Email: c_bagavathi@cb.amrita.edu

Intrusion detection systems (IDS) are crucial for network security, detecting and preventing unauthorized activities. This paper examines the effectiveness of IDS like Snort, Suricata, and Bro in analyzing network traffic and identifying anomalies across various application layer protocols such as DNS, SSH, FTP, SMTP, SNMP, and HTTPS. Each protocol poses unique challenges due to specific vulnerabilities, requiring IDS to utilize a mix of behavioral analysis, signature-based detection, and content inspection. Advanced techniques are essential for handling encrypted traffic in HTTPS and identifying threats in SMTP and DNS communications. The paper compares different IDS types—Network-Based, Host-Based, Protocol-Based, Application Protocol-Based, and Hybrid IDS—emphasizing the specialized protection offered by APIDS for application layer protocols. The integration of multiple IDS types enhances defense capabilities, underscoring the effectiveness of hybrid approaches for comprehensive threat management.

Keywords: Intrusion Detection System, Cybersecurity, Cyber-attacks.

2025. In Chetan Sharma, Vaishali Maheshwari & Harish Sharma (eds.), *Advancements in Intelligent Systems*, 45–69. Computing & Intelligent Systems, SCRS, India. DOI: https://doi.org/10.56155/978-81-975670-3-2-5

1 Introduction

In today's digital age where technology is pervasive, the demand for safety and security has surged significantly. This escalation is primarily driven by the constant evolution of cyber threats that pose risks to individuals, organizations, and nations alike. To effectively combat these ever-evolving cyber threats, there is a critical need for robust and reliable Intrusion Detection Systems (IDS). An Intrusion Detection System is a critical cybersecurity tool designed to monitor and analyze network traffic or system activities for signs of malicious activities or policy violations. It helps organizations detect and respond to cyber threats in real-time, enhancing their overall security posture [47].

One of the key areas of focus in cybersecurity is securing both the network and application layers. Safeguarding these layers is vital for ensuring overall cyber resilience and preserving the integrity of enterprise systems. However, in this paper, our primary emphasis lies on the Application protocol-based intrusion detection system (APIDS). APIDS delves into the intricate vulnerabilities associated with each application protocol, shedding light on how these vulnerabilities can be exploited by malicious actors.

Furthermore, this paper aims to explore the nuanced differences between APIDS and other protocol-based intrusion detection systems. Understanding these nuances is crucial for developing more effective cybersecurity strategies tailored to the specific challenges posed by application layer vulnerabilities.

This paper aims to explore how APIDS enhances cyber defenses in the application layer, bridging theory with practical application to contribute significantly to cybersecurity discussions.

2 APIDS Architecture and Protocol Techniques

Application Protocol-based Intrusion Detection Systems (APIDS) are designed to address vulnerabilities at the application layer by continuously monitoring the application-specific protocols. APIDS starts initiating its process by analyzing network traffic to identify the specific application layer protocol utilized by communicating hosts. Once a protocol is determined, APIDS customizes its intrusion detection system (IDS) response, focusing on the properties of the detected protocol. This initial analysis enables APIDS to adapt its detection mechanisms to match the characteristics of the identified application layer protocol, effectively detecting and mitigating threats unique to

that protocol [43].

2.1 Common Application protocols

The Application layer of the OSI (Open Systems Interconnection) model serves as the interface between the end user and the underlying network infrastructure. It encompasses a diverse range of protocols that facilitate various communication tasks, including data exchange, email transmission, web browsing, and remote access. Notable protocols include:

- **HTTP/HTTPS:** HTTP (Hyper Text Transfer Protocol) enables client-server communication for web browsing and data exchange, while HTTPS (Hyper Text Transfer Protocol Secured) enhances security through encrypted transmission between them [17].

- **SMTP:** Simple Mail Transfer Protocol(SMTP), facilitates email transmission between servers, defining the process of sending and receiving messages over the internet for seamless electronic mail exchange among users [8].

- **FTP/SSH:** FTP (File Transfer Protocol) and SSH (Secure Shell) are essential protocols used for transferring files and securing remote system access [51].

- **DNS:** DNS, the Domain Name System, decentralizes internet naming by translating domain names to IP addresses for seamless connectivity [12].

2.2 APIDS Architecture and Protocol Techniques

In Figure 1, various application protocols are depicted across four scenarios. When a user initiates a request to the internet, the system typically begins by querying the Domain Name System (DNS) if the request is an HTTP one. The DNS, responsible for translating domain names into IP addresses, facilitates this process. If the requested domain information is readily available in the DNS cache, or if it's stored within the DNS, the resolver retrieves the corresponding IP address and sends it back to the host [50]. However, if the information isn't cached, the resolver recursively queries other DNS servers until it successfully resolves the domain name, providing the IP address to the host.

Similarly, other protocols such as SMTP and FTP handle their requests and responses differently, with SMTP managing email exchange between servers and FTP facilitating file sharing within a network [38] [55].

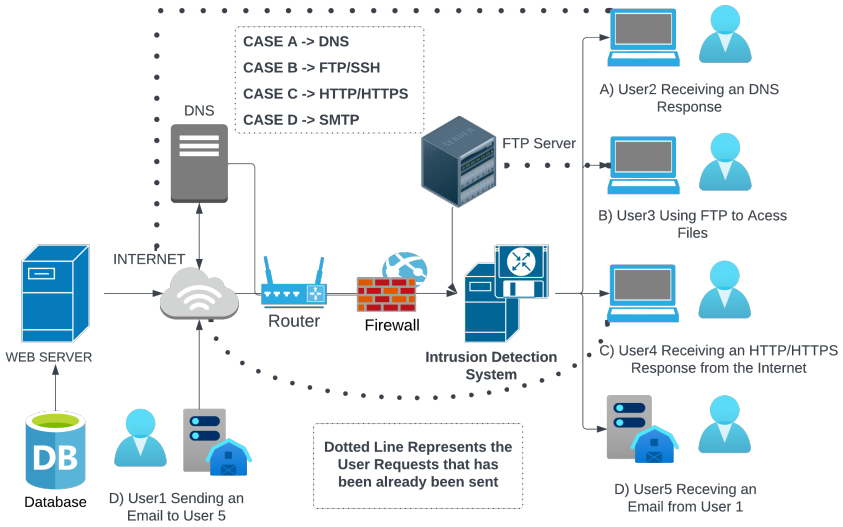

Figure 1: APIDS Architecture

Following the transmission of responses from each protocol, the intrusion detection system (APIDS) plays a crucial role in safeguarding against potential attacks. Typically positioned after the firewall at the receiving end, the APIDS employs protocol-specific techniques that are used in the Network-based Intrusion Detection System (NIDS).

These techniques enable the APIDS to identify the type of protocol being utilized (e.g., HTTP, FTP, SMTP) and analyze patterns and signature behaviors to detect intrusions specific to that protocol, which will be discussed below.

3 Protocol Identification Techniques

In this section, one of the important aspects of APIDS will be dealt with, which involves leveraging Protocol Techniques utilized in Network Intrusion Detection Systems (NIDS), particularly focusing on the ”Classification of network traffic”.

The ability to differentiate and classify network traffic is essential to iden-

tify potential risks and anomalies in the network. Over time, many methods have been developed and modified to accomplish this goal, each with its own unique and effective methods.

The four main approaches that focus on traffic classification, each of which provides specific insights into the nature of networks and helps to identify corresponding protocols are explained. These approaches include: i) classification based on port number ii) statistical analysis of traffic in a connection iii) locating specific protocol byte pattern in connection's payloads iv) dynamic Application-level protocol analysis.

3.1 Port Based Classification

Port-based classification, a foundational method in network traffic analysis, relies on port numbers to infer the associated protocol or service. However, servers may not always adhere to traditional port assignments due to benign or malicious reasons.

In networking, servers typically communicate on specific ports according to well-known protocols. However, they may deviate due to security, network configurations, or optimization. For example, consider a cloud storage service like Dropbox. While it primarily uses port 443 for secure file transfers (HTTPS), it might also utilize port 80 to ensure seamless access for users behind restrictive firewalls [25].

Similarly, when accessing a remote server via SSH, it conventionally operates on port 22. However, in environments where port 22 is blocked or heavily monitored, tools like GitLab might configure SSH to operate on port 443 instead, ensuring developers can securely push and pull code from repositories without hindrance [29].

Port multiplexing is also evident in popular web applications. For instance, a web server hosting a messaging platform like Slack might use port 443 for both HTTP and HTTPS traffic. This consolidation optimizes resource usage while maintaining secure communication channels for users [53].

Attackers exploit non-standard port usage to bypass security measures and gain unauthorized access. Using well-known ports like 80 or 443 for non-standard services bypasses firewall rules. Port multiplexing confuses network monitoring tools, providing opportunities for attackers to hide malicious activities. Overall, these tactics help attackers evade detection and establish unauthorized connections.

Table 1 shows the matched protocols and ports of data received during a research [24]. The research also suggested the 40% of packets received had mismatched port and protocol.

Table 1: Port based Protocol identification

Port	*HTTP*	*IRC*	*FTP*	*SMTP*	*Other*
80	92.2M	59	0	0	41.1K
6665-6669	1.2K	71.7K	0	0	4.2K
21	0	0	98.0K	2	2.3K
25	459	2	749	1.4M	195

3.2 Statistical analysis of traffic within a connection

Statistical analysis examines the characteristics of network traffic within individual connections to identify the associated protocol or service. This involves capturing and analyzing various attributes of network traffic, such as packet size, delays, and payload characteristics. By computing statistical metrics and patterns from these attributes, it becomes possible to differentiate normal traffic behavior from anomalous or potentially malicious activity.

Previous works [44] have used an analysis of interpacket delays and packet size distribution to differentiate interactive applications from bulk transfer applications [62]. Statistical analysis have yielded high accuracy scenarios [23] to differentiate web chat from web surfing.

A recent research [56] demonstrated the use of statistical connection analysis effectiveness of decision trees and neural networks in connection recognition, achieving a 90% average recognition rate. The neural engine's adaptability enables handling variations in network behavior, while its efficient resource usage supports real-time applications. Further enhancements are anticipated to improve accuracy and extend applicability to diverse data sources.

Other methods in statistical analysis includes creating profiles of normal network behavior based on statistical features observed over time. These profiles serve as reference models against which incoming traffic is compared. Deviations from established profiles can indicate potential intrusions or anomalies. Statistical analysis can also be used to detect anomalies, where deviations from expected statistical distributions are flagged as potential threats. Anomalies may manifest as sudden spikes or drops in traffic volume, unusual packet sizes, or unexpected patterns in packet inter-arrival times [11].

Analyzing traffic statistically can be complex, especially when dealing with large volumes of network data. Hackers might exploit this complexity by obfuscating their malicious activities within legitimate traffic, making it more difficult for intrusion detection systems to differentiate between normal and malicious behavior [59]. While statistical analysis of traffic within a connec-

tion offers valuable insights into network behavior and potential security threats, it is essential to recognize its limitations and vulnerabilities.

3.3 Locating protocol-specific byte patterns in the connection's payload.

Deep packet inspection techniques [3] can be employed to analyze the payload of network packets for protocol-specific byte patterns. Unlike port-based classification or statistical analysis, which focuses on header information or statistical features, this approach delves into the actual content of the packets to identify specific protocols or application layer protocols. Locating protocol-specific byte patterns involves scanning the payload of network packets for sequences of bytes that are characteristic of particular protocols or applications. This process requires knowledge of the protocol's data format, including header structures, message formats, and unique identifiers.

Signature-based detection involves predefining signatures or patterns associated with known protocols or applications. These signatures are then used to search the payload of network packets for matches, indicating the presence of the corresponding protocol or application. Regular expressions are powerful tools for specifying complex byte patterns or sequences. IDS can utilize regular expressions to define protocol-specific patterns and efficiently search packet payloads for matches.

By combining methods, we can utilize statistical approaches to cluster connections, then extract signatures, by machine learning techniques [31]. Alternatively, statistical methods may identify some applications, while signatures are used for others [62].

A recent research [24] performed using open source collection of application signatures included with the l7-filter system [1] and open source signature matching engine NIDS Bro [42]. Signatures are converted into Bro's syntax which gives advantages of Bro's trace processing, connection-oriented analysis, and powerful signature matching engine. Upon reviewing the results, it is clear that a single connection may occasionally activate multiple signatures. It uncovered that certain signatures within the l7-filter exhibit overly broad characteristics. For instance, the Finger protocol signature triggers are based solely on the presence of printable characters in the connection's initial two characters.

The use of static signatures or byte patterns can lead to false positives, where legitimate network traffic is incorrectly flagged as malicious. This can occur due to similarities between legitimate and malicious byte patterns or the presence of protocol-specific byte sequences in non-standard contexts.

Deep packet inspection techniques incur a performance overhead, particularly when analyzing large volumes of network traffic in real-time. The processing required to scan packet payloads for protocol-specific byte patterns may impact the throughput and latency of the intrusion detection system, potentially affecting its effectiveness in high-speed networks.

3.4 Dynamic Application-Layer Protocol Analysis

Dynamic Application-Layer Protocol Analysis relies on the real-time inspection of network traffic payloads to detect and analyze the behavior of different application-layer protocols. A processing path is included in this technique to dynamically add and remove analysis components. This technique relies on a per-connection data structure to represent the data path, which monitors the system's acquired knowledge regarding the analysis to conduct for the flow. For example, if the payload of a packet on port 80 initially analyzed as HTTP looks like an IRC session instead, we replace the HTTP analysis with IRC analysis.

This flexibility is achieved by associating a tree structure (where each node represents an analyzer) with the connection. To reduce the complexity of the tree the analyzer tree of a new connection only contains those analyzers definitely needed. Protocol Identification Analyzer(PIA) is used to match the protocol dynamically by adding and removing matched protocols [46].

A previous work [24] had been done to develop this technique. Open Source Bro was used to integrate PIA, and analyzer trees. A prediction table was implemented for storing the anticipated future connections. While the analyzer performed well compared to other techniques for identifying the protocols on non-standard ports, there is a certain amount of connections that are rejected in standard port. In Table 2 Column 2 shows the number of connections that are in non-standard ports but detected correctly by the analyzer. The third column tell about the connections which are not in standard port and not detected by the analyzer. The last column tells about the number of connections that are in the standard port but not detected by the analyzer.

Implementing dynamic analysis techniques for application layer protocols can be complex and resource-intensive. The process involves parsing and interpreting the content of network packets in real time, which may require significant computational resources and expertise. This technique may impose a heavy resource burden particularly when analyzing large volumes of network traffic. This can lead to performance degradation and scalability issues, especially in high-speed networks.

While port-based classification offers simplicity, it's vulnerable to evasion.

Table 2: Performance of Dynamic Protocol Identification

	Detected and verified non-std. port	***Rejected by analyzer non-std. port***	***Rejected by analyzer std. port***
HTTP	12,83,132	21,153	1,46,202
FTP	14,488	180	1,792
IRC	1,421	91	3
SMTP	69	0	1,368

Statistical analysis provides deeper insights but struggles with adaptability. Locating protocol-specific byte patterns offers granularity but lacks resilience to evasion. Dynamic Application-Layer Protocol Analysis presents promise, despite the complexity, offering adaptability crucial for real-time threat detection.

4 IDS for different application protocol

Network security relies heavily on intrusion detection systems (IDS), which are essential for spotting and stopping harmful or unauthorized activity. Organizations frequently utilize intrusion detection systems (IDS) to identify malicious network activity and defend against cyberattacks. Their efficiency can vary based on the environment and the models and technology used in their creation. Snort, Suricata, and Bro are three open-source intrusion detection systems that are widely used. Created in 1998, Snort is the most extensively used and researched intrusion detection system. The Open Information Security Foundation (OISF) created Suricata, a multi-threaded architecture for network traffic analysis. To identify anomalous network activity, Bro specifies distinct assaults in terms of events and blends signature- and anomaly-based detection techniques. [58]

The protocols covered in this subsection are those of the application layer, which includes DNS (Domain Name System), SSH (Secure Shell), FTP (File Transfer Protocol), SMTP (Simple Mail Transfer Protocol), SNMP (Simple Network Management Protocol), HTTPS (Hypertext Transfer Protocol Secure), and others. Because of its unique features and potential weaknesses, each of these protocols poses unique issues for intrusion detection. Network application communication is facilitated by application layer protocols. They comprise protocols like HTTP, FTP, and SNMP and function at the top tier

of the OSI model. IDS uses a set of rules to inspect packets that may include malicious content. An alert is generated if one of the rules is met by the packet payload. Examining application layer protocols for any questionable activity is part of this.

Testing IDS systems against a variety of attack types, such as DoS, DNS, FTP, scan port, and SNMP attacks, is necessary to assess their accuracy and performance. These assaults have the potential to impair IDS performance by focusing on particular application layer protocols. IDS performance is essential for managing the growing amount of network traffic. It should provide optimal accuracy with the fewest false positives and false negatives, as well as strong performance with no packet loss during analysis. [58]

4.1 HTTP/HTTPS

Web surfing and data transmission are prominent uses for HTTP (Hypertext Transfer Protocol), which makes communication between a client and server easier. A secure variant of HTTP, known as HTTPS (Hypertext Transfer Protocol Secure), guarantees encrypted and secure communication between the client and server.

Intrusion Detection Systems (IDS) are used in HTTP/HTTPS protocols to scan online traffic for unusual requests, including those involving SQL injection, cross-site scripting (XSS), DoS/DDoS attacks, directory traversal, and other web application threats. They identify irregularities in payloads and HTTP headers. Thus, HTTP intrusion detection systems need to be skilled at identifying and thwarting these kinds of attacks. HTTP traffic is mostly composed of client-server interactions related to data transmission and web browsing. As such, detection systems that can distinguish between malicious efforts to exploit vulnerabilities in online applications and authorized user activities are required. [35] [52]

Researchers create sophisticated models of protocols like HTTP using hybrid and heavy-tailed modeling techniques to increase the realism of testing environments. Session arrivals, bytes exchanged, and idle durations are all included in these models, which are essential for comprehending protocol behavior and enhancing IDS performance in threat detection. [39]

46% of all online web apps have serious vulnerabilities, while 87% have medium vulnerabilities, according to one report. Dangerous web cyberattacks surged by 300% alone in 2021. SQL injections, cross-site scripting, cache bypassing, and SYN floods are a few frequent HTTP and online threats. These flaws have the potential to seriously jeopardize web servers and online applications. [9]

Attackers mostly take advantage of weaknesses in web applications through the HTTP/HTTPS protocols. Because HTTPS encryption obscures network-based detection methods, intrusion detection systems (IDS) that detect HTTPS traffic are at a major disadvantage. Network packets are tracked by NIDS in order to identify and stop attacks. On the other hand, because HTTPS connection uses encrypted packet data, the system is unable to examine the contents of the packets. If an intrusion detection system (IDS) has access to the SSL certificate's private key, it can examine HTTPS traffic. The intrusion detection system (IDS) can examine encrypted HTTPS traffic for any security risks by decrypting it using the private key. Because HIDS are installed on the endpoints where encrypted data is decrypted and returned to its original form, they are capable of handling HTTPS traffic. [6]

4.2 SMTP

Email messages are sent between servers via the communication technology known as SMTP (Simple Mail Transfer technology). It establishes the parameters for sending and receiving emails via the internet, facilitating user-to-user email communication. Because of the growing significance of email, SMTP assaults and spam are major problems for administrators and users alike. SMTP encounters issues with email spoofing, phishing, and spamming. Additionally, SMTP is susceptible to malware propagation, buffer overflows, denial-of-service attacks, and other assaults that take advantage of weaknesses in open email systems. Strong intrusion detection systems are necessary in SMTP contexts, as demonstrated by a number of additional threats such as buffer overruns and partial message attacks. [20] [21]

SMTP intrusion detection systems need to distinguish between malicious activity and genuine email traffic by looking at email attachments, headers, content, and sender/receiver details. Behavioral analysis is used to monitor SMTP activity for unusual patterns over time. Signature-based detection is used to match known patterns of SMTP attacks. Anomaly-based detection is used to spot deviations in email traffic behavior. Content inspection is used to specifically scan email content for malware. Protocol compliance checks are used to ensure adherence to SMTP protocol standards. Finally, integration with threat intelligence feeds tailored to SMTP threats is employed. To identify and filter threats, lessen server load, and enhance performance, intrusion detection systems should be installed before email servers. IDS improves the overall security posture of the email infrastructure by assisting in the real-time identification and mitigation of security risks. [20] [21] [60]

Capturing and analyzing SMTP traffic can be very useful to improve the

identification of email-based threats. For example, using SMTP packet sniffing to record, preserve, and display email exchanges can greatly enhance intrusion detection capabilities. TCP streams and SMTP commands can be put back together by working on pcap files that contain SMTP traffic packets from an actual network, keeping the email headers that are recorded in a database. This method makes security management chores easier to comprehend and keep an eye on when it comes to SMTP transactions. By more intelligently evaluating email content, such a system can assist detect risks like malware and spam, hence enhancing network security. [7]

4.3 SNMP

In order to monitor and control network performance, devices can share management information more easily thanks to the Simple Network Management Protocol, or SNMP. Through the integration of distributed stateful intrusion detection into its centralized management architecture, the ID-Trace Management Platform expands the capabilities of SNMP. By transferring security responsibilities from management stations to mid-level managers, this platform makes proactive security measures possible. However, obstacles including difficult configuration procedures, problems integrating IDS with current network management systems, and restrictions on upgrading attack signatures stand in the way of IDS's widespread use. SNMP-based Network Management Systems (NMSs) use SNMP MIB statistical data to connect with IDSs through SVM-based algorithms to identify and mitigate traffic flooding attacks such as DoS/DDoS and Internet Worms, hence improving security. Correlation feature selection (CFS) is one of the efficient feature selection techniques that is used to find critical SNMP MIB variables for precise SVM-based attack detection and classification. Network security is further strengthened by the integration of SNMP with IDS tools like MMC in techniques like IID-LAN, which effectively identify anomalies and intrusions. [28] [15] [16]

4.4 FTP and SSH

Essential protocols for file transfers and secure remote system access are FTP (File Transfer Protocol) and SSH (Secure Shell), respectively. They can, however, be exploited; SSH is vulnerable to dictionary attacks and man-in-the-middle attacks, while FTP is targeted for brute-force attacks and illegal access. One of the most important functions of intrusion detection systems (IDS) is to protect these protocols. [10] [34] [33]

Attacks specific to FTP exist, such as FTP SITE EXEC and FTP bounce

attacks. IDS keeps an eye out for anomalies including odd file transfers, illegal access attempts, and FTP bounce assaults. It uses anomaly-based detection to find departures from typical FTP activity and signature-based detection to compare network data against known attack patterns. IDS also examines FTP traffic for indications of intrusion using behavioral and protocol analysis. It is possible to create FTP-specific IDS rules that will help identify potentially dangerous or suspicious activity. For example, a rule may be set up to send out a notification when it detects FTP connectivity. [19] [32]

Secure Shell (SSH) is a widely used standard for safeguarding data over the Internet. It is a secure login tool. Traffic from protocols like SSH is encrypted, which makes it challenging for IDS to examine and identify attacks. IDS keeps an eye out for brute-force attacks, illegal login attempts, and inconsistent protocol versions. Even though SSH encrypts communication, intrusion detection systems (IDS) use methods including flow time series analysis, traffic analysis of encrypted streams, and Hidden Markov Models (HMMs) to identify unusual activity suggestive of SSH attacks. IDS protects SSH connections by examining the SSH handshake, authentication attempts, and command executions. [27] [54]

Together, protocol analysis, usage pattern monitoring, anomaly-based detection, signature-based detection, and protocol analysis are used by IDS for FTP and SSH to examine FTP and SSH traffic, identify known attack patterns, and highlight deviations from normal behavior. When an intrusion is detected, intrusion detection systems (IDS) initiate reaction protocols, which may include notifications, connection termination, or access restriction implementation to lessen potential risks. IDS, in general, plays a vital defensive role in shielding SSH and FTP from malevolent use, guaranteeing the security and integrity of file transfers and remote system access. [10] [34] [52]

4.5 DNS

Domain Name System, or DNS, is a decentralized naming system that converts domain names into IP addresses for computers, services, and other internet-connected entities. Attackers target DNS in an attempt to compromise network security, divert traffic, or pilfer private data. Its lack of built-in security mechanisms and hierarchical structure make it simple to alter, spoof, and exploit flaws. By closely examining DNS traffic for different threats, IDS is essential in reducing the impact of DNS attacks. It keeps an eye on network activity and interacts with DNS servers to identify and stop threats like DNS amplification and tunneling. IDS systems with machine learning as its foundation provide very accurate DNS DoS attack detection. IDS signatures are

designed to identify particular DNS threats, such as amplification and tunneling attacks. IDS identifies illegal actions such as command and control using DNS infrastructure and faking victim IP addresses by continually monitoring DNS traffic. It improves the overall security of DNS infrastructure by facilitating the early identification and mitigation of DNS assaults. By using techniques like analyzing DNS channel volume and packet size characteristics, intrusion detection systems (IDS) block malicious DNS packets and domains associated with known threats and notify administrators of threats that need to be addressed immediately. Zeek, the powerful network analysis platform that replaces Bro-IDS, collects and stores copious amounts of network traffic data, including DNS logs. Bro-IDS extracts DNS data and offers details on visited domains and originating hosts. It enhances cybersecurity and monitors for anomalies in DNS traffic by detecting suspicious activities. [5] [45] [38] [50] [22]

Explainable AI (XAI) techniques, like SHAP, can improve transparency in IDS decision-making and help to detect DoH attacks effectively. The EfficientIP and IDC 2021 Global DNS Threat Report states that approximately 87% of surveyed organizations experienced DNS attacks in 2021, highlighting the need for robust IDS solutions. With the rise of DNS over HTTPS (DoH), which encrypts DNS queries within HTTPS, IDS faces new challenges in analyzing encrypted traffic. [61] [38]

5 Comparision and effectiveness of apids with other IDS

IDS are generally Categorized into 5 types according to the National Institute of Standards and Technology (NIST) Scarfone and Mell (2007) and they are i) NIDS (Network Based IDS) ii) HIDS (Host Based IDS) iii) PIDS (Protocol Based IDS) iv) Hybrid Based IDS and we have our APIDS.

1. ***NIDS*** - Observing network traffic involves monitoring the data packets that traverse a network to ensure its security, performance, and reliability. This process is essential in identifying potential security threats, performance bottlenecks, and network-related issues.

2. ***HIDS*** - Monitors the Incoming and Outgoing of a particular Host in the network by comparing the Previous and the Current State of the system.

3. ***PIDS*** - Monitor and intercept network packets associated with any protocol across all layers of the OSI model, detecting anomalies and potential security threats.

4. ***APIDS*** - APIDS can monitor and intercept packets specific to application-layer protocols in the OSI model, analyzing traffic to detect and prevent attacks.

5. ***Hybrid*** - It is a combination of two or more of the above IDS to form a strong Protection which is more efficient compared to the Tradional Approaches.

The Clear comparison of Advantages and Disadvantages is shown in the tabular in Table 3. The Table gives the comparison of IDS like APIDS, HIDS, NIDS, WIDS(Wireless Intrusion Detection Systems), MIDS(Microwave Intrusion detection systems). While this paper focuses majorly on APIDS the other IDS can be derived through other researches and surveys [41].

In evaluating APIDS against other IDS variants, it excels in pinpointing vulnerabilities within application layer protocols, ensuring focused security measures. While APIDS stands out for its specialized protection, the synergistic integration of various IDS types, such as NIDS, HIDS, or PIDS, enhances overall defense capabilities, highlighting the efficacy of hybrid approaches for comprehensive threat management. [14] [2]

6 Future Directions in APIDS Research

To increase threat detection efficiency across both standard and non-standard ports, future research in Application Protocol-based Intrusion Detection Systems (APIDS) should concentrate on improving protocol analysis approaches, especially for real-time packet parsing and interpretation [30]. Through adaptive pattern analysis-based threat recognition, machine learning integration can further improve APIDS [13]. Concerning protocols such as HTTPS, it is particularly important to address the problem of analyzing encrypted traffic without sacrificing privacy. [26] [36] [57]

Additionally, optimizing the performance of APIDS is vital to maintaining its effectiveness in high-speed network environments. Given the resource-intensive nature of deep packet inspection, future research should focus on minimizing latency and maximizing throughput, ensuring that APIDS can handle large volumes of traffic without degrading network performance [26].

Table 3: Advantages and Disadvantages of various IDS

IDS	Advantages	Disadvantages
NIDS	Independent of Operating Environments.	Does not indicate whether the attack was successful or not. Cannot Analyze Encrypted Traffic. NIDS is has very limited visibility inside the host machine.
HIDS	HIDS can analyze encrypted data and communications activity. HIDS can identify if an attack is successful or not. Easy to deploy because it does not require additional hardware, therefore, it does not affect the current architecture.	Does not indicate whether the attack was successful or not. Cannot Analyze Encrypted Traffic. NIDS is has very limited visibility inside the host machine.
MIDS	More flexible. More Efficient. MIDS take advantage of the strengths of the combined type.	High overhead load on the monitored system depending on the combined methodologies. Processor utilization of the hybrid agent is much great.
WIDS	More accurate. It can manage wireless protocol activity.	Sensors has limited computational resource and limited energy. [18]
APIDS	APIDS focus on observing and analyzing operations particular to the application. More easier to define the normal and the abnormal behavior.	Larger system overhead. Specific development. [37] It does not detect attacks below the application layer.

Aside from focusing on performance optimization to manage high-speed networks with minimal latency, efforts should also be focused on automating signature creation to quickly respond to new threats. To ensure strong defenses against constantly emerging cyber threats, a more comprehensive security framework can be created by integrating APIDS with other IDS types and extending protocol coverage. [48] [51]

In addition to these areas, future research could also focus on enhancing APIDS integration with broader cybersecurity systems, such as Security Information and Event Management (SIEM) platforms. This would enable more context-aware detection and facilitate real-time threat correlation across different layers of an organization's security infrastructure [40] [4]. By incorporating threat intelligence feeds, APIDS could become more proactive in identifying potential threats, contributing to a more resilient and adaptive cybersecurity strategy. These advancements are crucial for keeping pace with the rapidly evolving cyber threat landscape, ensuring that APIDS remains a critical component of comprehensive network defense mechanisms.

7 Conclusion

In conclusion, the deployment of Application Protocol-based Intrusion Detection Systems (APIDS) is becoming increasingly essential for enhancing modern network security due to their unique focus on monitoring application-layer traffic. Unlike traditional intrusion detection systems that concentrate on network or host-level activities, APIDS are specifically designed to analyze application protocols such as HTTP, HTTPS, SMTP, SNMP, FTP, SSH, and DNS. This targeted approach enables APIDS to effectively detect and mitigate complex attacks that exploit vulnerabilities within these protocols—threats that might not be identified by other types of IDS.

By providing deep visibility into application-layer traffic, APIDS can detect anomalies and patterns indicative of malicious activity, such as SQL injection, cross-site scripting, and other protocol-specific attacks. This capability is crucial in identifying and responding to threats that target the application layer directly. Furthermore, APIDS can seamlessly integrate with existing network infrastructure, enhancing overall security posture without causing significant disruptions especially in the domain of Industrial IoT and similar state-of-the-art applications [49].

As cyber threats continue to evolve and become more targeted, the ability of APIDS to adapt and respond to emerging vulnerabilities becomes increasingly critical. Their focused approach on application protocols ensures

that organizations are better prepared to defend against new and advanced threats. Therefore, the adoption of APIDS is essential for maintaining a robust and resilient network defense strategy, capable of proactively detecting and mitigating potential threats at the application layer.

References

[1] Application layer packet classifier for linux.

[2] Evaluation of machine learning algorithms for intrusion detection utilizing unsw-nb15 dataset 2022. pages 4819–4832, Dec. 2022.

[3] Tamer Abuhmed, Abedelaziz Mohaisen, and Daehun Nyang. Deep packet inspection for intrusion detection systems: A survey. 2007.

[4] Jehad Monzer Abuneama, Mohammed AI Matar, and Aiman Ahmed Abusamra. Enhancing cybersecurity with ids and siem integration detection check for updates. *AI in Business: Opportunities and Limitations*, 2:57, 2024.

[5] Sanjay Adiwal, Balaji Rajendran, Pushparaj Shetty D., and Sithu D. Sudarsan. Dns intrusion detection (did) — a snort-based solution to detect dns amplification and dns tunneling attacks. *Franklin Open*, 2:100010, 2023.

[6] Nancy Agarwal and Syed Zeeshan Hussain. A closer look on intrusion detection system for web applications. *CoRR*, abs/1803.06153, 2018.

[7] Maurizio Aiello, David Avanzini, Davide Chiarella, and Gianluca Papaleo. Smtp sniffing for intrusion detection purposes. 01 2007.

[8] Jamal N Al-Karaki, Amjad Gawanmeh, and Claude Fachkha. Blockchain for email security: A perspective on existing and potential solutions for phishing attacks. In *2023 Fifth International Conference on Blockchain Computing and Applications (BCCA)*, pages 404–411. IEEE, 2023.

[9] Hani M. Al-Mimi, Nesreen A. Hamad, and Mosleh M. Abualhaj. A model for the disclosure of probe attacks based on the utilization of machine learning algorithms. *2023 10th International Conference on Electrical and Electronics Engineering (ICEEE)*, pages 241–247, 2023.

[10] Noura Alotibi and Majid Alshammari. Deep learning-based intrusion detection: A novel approach for identifying brute-force attacks on ftp

2025. In Chetan Sharma, Vaishali Maheshwari & Harish Sharma (eds.), *Advancements in Intelligent Systems*, 45–69. Computing & Intelligent Systems, SCRS, India. DOI: https://doi.org/10.56155/978-81-975670-3-2-5

and ssh protocol. *International Journal of Advanced Computer Science and Applications*, 14(6), 2023.

[11] Ankitdeshpandey and R Karthi. Development of intrusion detection system using deep learning for classifying attacks in power systems. In *Soft Computing: Theories and Applications: Proceedings of SoCTA 2019*, pages 755–766. Springer, 2020.

[12] Ibrahim Ayoub, Sandoche Balakrichenan, Kinda Khawam, and Benoît Ampeau. Dns for iot: a survey. *Sensors*, 23(9):4473, 2023.

[13] Zahedi Azam, Md Motaharul Islam, and Mohammad Nurul Huda. Comparative analysis of intrusion detection systems and machine learning based model analysis through decision tree. *IEEE Access*, 2023.

[14] Youakim Badr. Enabling intrusion detection systems with dueling double deep q-learning. *Digital Transformation and Society*, 1(1):115–141, August 2022. Publisher Copyright: © 2022, Youakim Badr.

[15] Cui-Mei Bao. Intrusion detection based on one-class svm and snmp mib data. In *2009 Fifth International Conference on Information Assurance and Security*, volume 2, pages 346–349, 2009.

[16] Cui-Mei Bao. Intrusion detection based on one-class svm and snmp mib data. In *2009 Fifth International Conference on Information Assurance and Security*, volume 2, pages 346–349, 2009.

[17] Ezmolda Barolli, Loren Nebiaj, and Gloria Tyxhari. Data traffic and security over internet via monitoring and analyzing the http protocol. *International Journal of Engineering & Technology IJET-IJENS*, 14(16):44–50, 2014.

[18] O. Can and O. K. Sahingoz. A survey of intrusion detection systems in wireless sensor networks. In *Modeling, Simulation, and Applied Optimization (ICMSAO), 2015 6th International Conference on*. IEEE, 2015.

[19] Chee Chan and Alexander Yeoh. Development of a platform to explore network intrusion detection system (nids) for cybersecurity. *Journal of Computer and Communications*, 06:1–11, 01 2018.

[20] Ming-Jen Chen, Kuan-Ping Chien, Chia-Ying Huang, Bo-Chao Cheng, and Yuan-Sun Chu. An asic for smtp intrusion prevention system. *2009 IEEE International Symposium on Circuits and Systems*, pages 1847–1850, 2009.

[21] Bo-Chao Cheng, Ming-Jen Chen, Yuan-Sun Chu, Andrew Chen, Sujadi Yap, and Kuo-Pao Fan. Sips: A stateful and flow-based intrusion prevention system for email applications. In *IFIP International Conference on Network and Parallel Computing*, 2007.

[22] Abdulla Dakhgan, Ali Hadi, Jaafer Al Saraireh, and Doaa Alrababah. Passive dns analysis using bro-ids. In *2017 International Conference on New Trends in Computing Sciences (ICTCS)*, pages 121–126, 2017.

[23] Christian Dewes, Arne Wichmann, and Anja Feldmann. An analysis of internet chat systems. In *Proceedings of the 3rd ACM SIGCOMM Conference on Internet Measurement*, IMC '03, page 51–64, New York, NY, USA, 2003. Association for Computing Machinery.

[24] Holger Dreger and Anja Feldmann. Dynamic application-layer protocol analysis for network intrusion detection. In *USENIX Security Symposium*, 2006.

[25] Dropbox. Configuring your firewall for dropbox. 2024. Accessed: 2024-07-11.

[26] Reham Taher El-Maghraby, Nada Mostafa Abd Elazim, and Ayman M. Bahaa-Eldin. A survey on deep packet inspection. In *2017 12th International Conference on Computer Engineering and Systems (ICCES)*, pages 188–197, 2017.

[27] Vahid Aghaei Foroushani, Fazlollah Adibnia, and Elham Hojati. Intrusion detection in encrypted accesses with ssh protocol to network public servers. *2008 International Conference on Computer and Communication Engineering*, pages 314–318, 2008.

[28] L.P. Gaspary, R.N. Sanchez, D.W. Antunes, and E. Meneghetti. A snmp-based platform for distributed stateful intrusion detection in enterprise networks. *IEEE Journal on Selected Areas in Communications*, 23(10):1973–1982, 2005.

[29] GitLab Documentation. *SSH Executors | GitLab*. GitLab Inc., 2024. Accessed: 2024-07-11.

[30] Vitor Graveto, Tiago Cruz, and Paulo Simões. A network intrusion detection system for building automation and control systems. *IEEE Access*, 11:7968–7983, 2023.

[31] Patrick Haffner, Subhabrata Sen, Oliver Spatscheck, and Dongmei Wang. Acas: automated construction of application signatures. In *Proceedings of the 2005 ACM SIGCOMM Workshop on Mining Network Data*, MineNet '05, page 197–202, New York, NY, USA, 2005. Association for Computing Machinery.

[32] Guy Helmer, Johnny Wong, Mark Slagell, Vasant Honavar, Les Miller, and Robyn Lutz. A software fault tree approach to requirements analysis of an intrusion detection system. *Requirements Engineering*, 7, 12 2000.

[33] Muhammad Agreindra Helmiawan, Eggi Julian, Yavan Cahyan, and Asep Saeppani. Experimental evaluation of security monitoring and notification on network intrusion detection system for server security. In *2021 9th International Conference on Cyber and IT Service Management (CITSM)*, pages 1–6, 2021.

[34] Md. Delwar Hossain, Hideya Ochiai, Doudou Fall, and Youki Kadobayashi. Ssh and ftp brute-force attacks detection in computer networks: Lstm and machine learning approaches. *2020 5th International Conference on Computer and Communication Systems (ICCCS)*, pages 491–497, 2020.

[35] Hossein Hadian Jazi, Hugo Gonzalez, Natalia Stakhanova, and Ali A. Ghorbani. Detecting http-based application layer dos attacks on web servers in the presence of sampling. *Computer Networks*, 121:25–36, 2017.

[36] Tiina Kovanen, Gil David, and Timo Hämäläinen. Survey: Intrusion detection systems in encrypted traffic. In Olga Galinina, Sergey Balandin, and Yevgeni Koucheryavy, editors, *Internet of Things, Smart Spaces, and Next Generation Networks and Systems*, pages 281–293, Cham, 2016. Springer International Publishing.

[37] Aleksandar Lazarevic, Vipin Kumar, and Jaideep Srivastava. Intrusion detection: A survey. In Vipin Kumar, Jaideep Srivastava, and Aleksandar Lazarevic, editors, *Managing Cyber Threats*, pages 19–78. Springer, 2005.

[38] Allan Liska and Geoffrey Stowe. *DNS Security: Defending the Domain Name System.* Syngress, 2016.

[39] Song Luo and G.A. Marin. Modeling networking protocols to test intrusion detection systems. pages 774–775, 2004.

[40] Adabi Raihan Muhammad, Parman Sukarno, and Aulia Arif Wardana. Integrated security information and event management (siem) with intrusion detection system (ids) for live analysis based on machine learning. *Procedia Computer Science*, 217:1406–1415, 2023.

[41] Suad Othman, Taha Nabeel, Fadl Ba-Alwi, and Ammar Zahary. Survey on intrusion detection system types. 12 2018.

[42] Vern Paxson. Bro: a system for detecting network intruders in real-time. *Computer Networks*, 31(23):2435–2463, 1999.

[43] Rama Prabha and KP Jayanthi R. Application-protocol based intrusion detection system. *International Journal of Computer Science & Management Studies*, 14(10), 2014.

[44] Mohammed Abdul Qadeer, Arshad Iqbal, Mohammad Zahid, and Misbahur Rahman Siddiqui. Network traffic analysis and intrusion detection using packet sniffer. pages 313–317, 2010.

[45] Cheng Qi, Xiaojun Chen, Cui Xu, Jinqiao Shi, and Peipeng Liu. A bigram based real time dns tunnel detection approach. *Procedia Computer Science*, 17:852–860, 12 2013.

[46] K Rahul-Vigneswaran, Prabaharan Poornachandran, and KP Soman. A compendium on network and host based intrusion detection systems. In *ICDSMLA 2019: Proceedings of the 1st International Conference on Data Science, Machine Learning and Applications*, pages 23–30. Springer, 2020.

[47] S Aravinth Raj, PP Amritha, Sethumadhavan, and Srinivasan Seshadhri. Hybrid intrusion detection system for industrial control system. In *World Conference on Information Systems for Business Management*, pages 573–583. Springer, 2023.

[48] Muhammad Sajid, Kaleem Razzaq Malik, Ahmad Almogren, Tauqeer Safdar Malik, Ali Haider Khan, Jawad Tanveer, and Ateeq Ur Rehman. Enhancing intrusion detection: a hybrid machine and deep learning approach. *Journal of Cloud Computing*, 13(1):123, 2024.

[49] V. Saranya, M. J. Carmel Mary Belinda, and G. R. Kanagachidambaresan. *An Evolution of Innovations Protocols and Recent Technology in Industrial IoT*, pages 161–175. Springer International Publishing, Cham, 2020.

[50] Pratik Satam, Hamid Alipour, Youssif Al-Nashif, and Salim Hariri. Dns-ids: Securing dns in the cloud era. In *2015 International Conference on Cloud and Autonomic Computing*, pages 296–301, 2015.

[51] Shishir Kumar Shandilya, Chirag Ganguli, Ivan Izonin, and Atulya Kumar Nagar. Cyber attack evaluation dataset for deep packet inspection and analysis. *Data in Brief*, 46:108771, 2023.

[52] Marek Sikora, Radek Fujdiak, and Jiri Misurec. Analysis and detection of application-independent slow denial of service cyber attacks. In *2021 IEEE International Conference on Intelligence and Security Informatics (ISI)*, pages 1–6, 2021.

[53] Slack Documentation. *How to Read Slack Data Exports*. Slack Technologies, 2024. Accessed: 2024-07-11.

[54] Anna Sperotto, M.R.H. Mandjes, R. Sadre, Pieter-Tjerk de Boer, and Aiko Pras. Autonomic parameter tuning of anomaly-based idss: an ssh case study. *IEEE transactions on network and service management*, 9(2):128–141, June 2012. eemcs-eprint-21923.

[55] William Stallings. *Network Security Essentials: Applications and Standards*. Pearson, 2016.

[56] K.M.C. Tan and B.S. Collie. Detection and classification of tcp/ip network services. In *Proceedings 13th Annual Computer Security Applications Conference*, pages 99–107, 1997.

[57] Ankit Thakkar and Ritika Lohiya. A review on challenges and future research directions for machine learning-based intrusion detection system. *Archives of Computational Methods in Engineering*, 30(7):4245–4269, 2023.

[58] Kittikhun Thongkanchorn, Sudsanguan Ngamsuriyaroj, and Vasaka Visoottiviseth. Evaluation studies of three intrusion detection systems under various attacks and rule sets. pages 1–4, 2013.

[59] Ravi Vinayakumar, Mamoun Alazab, K Padannayil Soman, Prabaharan Poornachandran, Ameer Al-Nemrat, and Sitalakshmi Venkatraman. Deep learning approach for intelligent intrusion detection system. *Ieee Access*, 7:41525–41550, 2019.

[60] Bailin Xie and Qiansheng Zhang. Application-layer anomaly detection based on application-layer protocols' keywords. In *Proceedings of 2012 2nd International Conference on Computer Science and Network Technology*, pages 2131–2135, 2012.

[61] Tahmina Zebin, Shahadate Rezvy, and Yuan Luo. An explainable ai-based intrusion detection system for dns over https (doh) attacks. *IEEE Transactions on Information Forensics and Security*, 17:2339–2349, 2022.

[62] Yin Zhang and Vern Paxson. Detecting backdoors. In *USENIX Security Symposium*, 2000.