International Scientific Journal of Engineering and Management (ISJEM) ISSN: 2583-6129
Volume: 04 Issue: 07 | July – 2025 DOI: 10.55041/ISJEM04702
An International Scholarly || Multidisciplinary || Open Access || Indexing in all major Database & Metadata

Evaluating an Intrusion Detection System for Wireless Sensor Networks Within the Internet of Things Using a Metrics Scorecard Method Focused on Architecture

Prabhjot Kaur [1], Rupinder Singh [2], Rachhpal Singh [3]

[123] Department of Computer Science, Khalsa College Amritsar,Punjab. E-mail: prabhjot25762@gmail.com

ABSTRACT- The expected range, the structure of IOT WSNIDS, and their relationship with the deployment architecture are all assessed using IOT WSNIDS architectural metrics. These metrics can be employed to gauge the efficiency of an IOT WSNIDS architecture and assist in creating effective IOT WSNIDS. IOT WSNIDS are crucial for ensuring the security of wireless sensor networks by monitoring wireless-specific traffic, including detecting external users attempting to connect to the network via access points. As wireless technology is continuously advancing, developing IOT WSNIDS presents a significant challenge. Architectural metrics can significantly influence the design of IoT Wireless Sensor Network Intrusion Detection Systems (WSNIDS) by identifying the problematic areas within its architecture. This study explores a range of architectural metrics that are relevant to IoT WSNIDS. The primary focus of testing and evaluating an IoT WSNIDS revolves around a "scorecard" that includes a set of values. An IoT WSNIDS can be appraised by assigning scores to various architectural metrics associated with it. We illustrate our architectural metrics scorecard-based evaluation approach using three prominent IoT WSNIDS: Snort, Suricata, and Zeek. In conclusion, we present the findings and highlight the significant potential for further research within this domain.

Keywords: Architectural Metrics, IOT WSN, Metrics, IDS, and Scorecard.

I. INTRODUCTION

IOT WSNIDS has opened up a remarkable new realm. Its technology is advancing daily, and its user base continues to grow. However, the primary concern with IOT WSNIDS has been related to security. For some time, IOT WSNIDS operated with minimal security measures, often in a highly exposed environment. The IOT WSN Intrusion Detection System offers a novel solution to address this problem, along with improved encryption techniques. An intrusion detection system (IDS) is a software or hardware tool that observes network and/or system activity for harmful actions or breaches of policy and produces reports for a management center (Wikipedia, 2012). This monitoring is specifically carried out for wireless networks by a wireless IDS. This technology tracks network traffic for potential vulnerabilities and notifies personnel to take necessary action.

"If you can't measure it, you can't enhance it," said Lord Kelvin. This principle is also relevant when discussing wireless network security. This widely recognized management theory applies to security as well; without measurement, management of an activity is impossible. Metrics can serve as a valuable resource for security providers to evaluate the effectiveness of different components within security programs. Metrics play a crucial role in the design of IoT WSNIDS. Given that the domain of wireless network security is still emerging, establishing security metrics for this technology proves challenging. There remains an absence of a standardized terminology and well-documented best practices [1].

To assess intrusion detection systems, which have gained popularity in the commercial sector for IoT Wireless Sensor Networks (WSN), this article introduces a methodology based on a scorecard of architectural metrics. We present a testing strategy designed to evaluate IoT WSN Intrusion Detection Systems (IDS) by assigning scores to various relevant architectural metrics. The methodology discussed in this study evaluates IoT WSN IDS against a set of architectural metrics pertinent to them, rather than comparing the systems with each other. Thanks to the generalized approach of this paper, systems that require wireless capabilities will be able to tailor their evaluation of IDS technologies to fit their specific needs. The evaluation can potentially be broadened to incorporate additional metrics such as logistical, performance, quality metrics, and more, since the assessment corresponds to a fixed set of architectural metrics. The standard comparison approach outlined in this paper also ensures scientific reproducibility.

II. SNORT, SURICATA AND ZEEK IDS

We chose three IOT WSNIDS—Snort, Suricata, and Zeek—as they are among the most well-known and utilize various technologies—in order to illustrate the architectural metrics scorecard based evaluation method to IOT WSNIDS.

(a) Snort

SNORT is a powerful open-source system for intrusion detection and prevention (IDS and IPS) that analyzes network traffic in real-time and tracks data packets. To identify potentially harmful activities, SNORT utilizes a rule-based language that combines anomaly detection, protocol analysis, and signature-based inspection methods.

Network administrators can identify threats such as Common Gateway Interface (CGI) attacks, buffer overflow incidents, stealth port scans, as well as denial-of-service (DoS) and distributed denial-of-service (DDoS) attacks through SNORT. Malicious network behaviors are characterized by a set of rules created by SNORT, which identify harmful packets and alert users.

SNORT is open-source software that can be used both personally and commercially. The SNORT rule language specifies what network traffic should be monitored and what actions should be taken when suspicious packets are encountered. This snorting functionality serves to detect unauthorized packets similarly to sniffers and network intrusion detection systems or functions as a comprehensive network IPS solution that monitors traffic and identifies as well as blocks potential attack vectors.

(b) Suricata

Suricata is a fantastic, inexpensive tool that provides deeper insights into network traffic. However, it should be regarded as just one aspect of a holistic security strategy, rather than a standalone fix for all security challenges. Snort is one of the most frequently used alternatives, and it has been popular among administrators for quite some time. While Suricata is relatively newer, it comes with several benefits. Although both tools operate on different architectures, they can utilize the same signatures. A significant distinction between the two is that Suricata operates in a multi-threaded manner, allowing it to leverage multiple cores simultaneously. By using several CPUs, Suricata can handle numerous events concurrently without interrupting other requests. Additionally, multi-threading allows for effective load balancing across CPUs and enhances the overall performance of network traffic analysis. This is beneficial as it enables Suricata to analyze substantial amounts of traffic without having to reduce the number of rules used.

The engine is built to leverage the latest multi-core CPU chipsets and make use of hardware acceleration for enhanced processing capabilities. Suricata's high efficiency, support for IP reputation, and automated protocol detection contribute to its effectiveness in providing better visibility into a network.

(c) Zeek

Zeek, is a free and open-source tool for analyzing network traffic. This software operates on a sensor to monitor network activity. It is designed to extract numerous fields from network data in real-time, all at no cost. Zeek includes pre-existing parsers for a variety of protocols, such as HTTP, SSL, DNS, and FTP, and it also supports the creation of custom parsers for unsupported protocols. While Zeek can identify anomalies, it does so differently compared to traditional Intrusion Detection Systems (IDS) like Suricata. The tool replicates (or SPANs) a router within your network to gather a duplicate of the traffic. It subsequently processes, analyzes, and organizes the network data according to protocols. The processed information is then saved into different log files (dns.log, http.log, con.log, etc.).

Accurate network data is essential when analyzing security incidents and developing effective detection methods. To gain a comprehensive understanding of the happenings within your network, it is crucial to recognize which systems and services are connecting and to analyze the flow of traffic across your organization's IT structure. There are numerous approaches to gather network data. You can obtain it through a firewall, netflow, or other network analysis tools and technologies. Nevertheless, this data is often either incomplete or prohibitively expensive to acquire. This is where Zeek comes into play. Zeek serves as an excellent resource for network data related to threat hunting, monitoring, and analysis. When set up properly, it doesn't overwhelm the network or burden security teams with unnecessary information. It extracts specific fields from network data to deliver parsed and actionable insights that can be utilized to develop effective detections for enhancing network security.

Table 1: **Selected Architectural metrics for IOT WSNIDS**

Architectural Metrics	Description
Adjustable Sensitivity	The difficulty of altering the sensitivity of a IOT WSNIDS in order to achieve a balance between false positive and false negative error rates at various times and for different environments.
Required Data Storage Capacity	The amount of disk space needed to store logs and other application data.
Load Balancing Scalability	It measures the ability of a IOT WSNIDS to partition traffic into independent, balanced sensor loads.
Multiple Sensor Support	The cardinality of sensors supported.
Reordering and Stream Reassembly	It can be used to find an attack that has been artificially fragmented and transmitted out of order.
State Tracking	This metric is useful in hardening IOT WSNIDS against storms of random traffic used to confuse it.
Data Pool Selectability	This metric is used to define the data source to be analyzed for intrusions.
System Throughput	Maximal data input rate that can be processed successfully by the IOT WSNIDS.

III AN APPROACH BASED ON SCORECARDS FOR ARCHITECTURAL METRICS

(a) Developing Scorecard

A "scorecard" featuring a compilation of architectural criteria and their explanations will act as the primary tool for assessing and evaluating IoT WSNIDS. Each metric can receive a score of low (+), moderate (++), or high (+++), with higher scores indicating more significant ratings. The architectural metrics considered encompass general characteristics relevant to the architecture of an IoT WSNIDS. To ascertain the value of each architectural metric, one can use both analysis (such as source code examination) and publicly available content (including specifications, white papers, or feedback from vendors or users). In our evaluation of each architectural metric for IoT WSNIDS, we rely on open-source resources. We delve into publicly available materials such as conference proceedings, research studies, reports, product manuals, and other documents that are accessible for public review.

(b) Architectural Metrics for a IOT WSNIDS

The expected framework and layout of the IoT WSNIDS are evaluated against the deployment architecture utilizing architectural metrics. These metrics measure the architectural efficiency of an IoT WSN IDS [15]. Table 1 presents the metrics identified in that field. Additional architectural metrics that may be utilized include Anomaly Based, Autonomous Learning, Host/OS Security, Interoperability, Package Contents, Process Security, Signature Based, and Visibility [7].

(c) Approach Based on an Architectural Metrics Scorecard

In this section, we will utilize the previously mentioned method to evaluate the well-known IoT WSNIDSs: Snort, Suricata, and Zeek. We chose to assess these three because they are among the most widely used and function differently. The scoring system for architectural metrics related to these three IoT WSNIDS is explained below, with reference to Table 2. Scores for the architectural metric
Adjustable Sensitivity can be established based on the following criteria:
Low Score (+): No ability to adjust.
Average Score (++): Adjustability via static methods.
High Score (+++): Intelligent, dynamic Adjustability.

Table 2 : Scorecard for Snort,Suricata, and Zeek IDS.

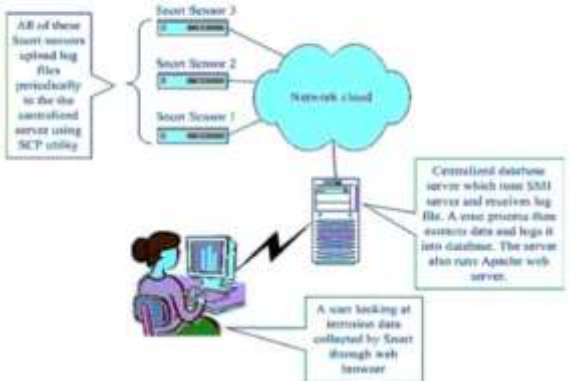
+ : Low score ; ++ : Average score ; +++ : High Score

Architectural Metrics	Snort	Suricata	Zeek
Adjustable Sensitivity	+++	++	++
Required data Storage Capacity	+	++	+++
Load Balancing Scalability	+++	+++	++
Multiple Sensor Support	+++	+++	+++
Reordering and Stream Reassembly	+++	+++	+++
State Tracking	+++	+++	+++
Data Pool Selectability	+++	+++	+++
System Throughput	+++	++	++

The SSL Dynamic Preprocessor (SSLPP), a part of Snort, facilitates the decoding of SSL and TLS traffic and decides when Snort should cease its inspection of such data. To enhance efficiency and minimize the chances of false positives and negatives, Snort disregards encrypted traffic. Consequently, Snort earns a high rating (+++) for the metric of adjustable sensitivity. Zeek sends alerts based on fingerprints, particularly for certain versions of nets tumbler. Nets tumbler broadcasts unique packets to try and disclose the SSID of a network. Although this is not always executed, when it occurs, the chance of false positives is very minimal. As a result, Zeek is assigned an average score for the metric of adjustable sensitivity. Suricata encountered a false positive during a Nets tumbler scan, which was actually one of the test laptops pinging an access point. Suricata indicates that improvements are necessary for the Nets tumbler signature, leading to an average rating for the metric of adjustable sensitivity.
The Required Data Storage Capacity for architectural metrics can be evaluated based on the following criteria: Low Score (+): Significant storage capacity is necessary for keeping logs and various files. Average Score (++): A medium level of storage capacity is needed for the logs and other files. High Score (+++): Minimal storage capacity is required for storing logs and additional files. Snort utilizes databases for storing log and alert data. For smaller systems, saving log data directly to disk files is sufficient. However, when dealing with multiple Snort sensors or the need to retain historical data, using disk files for log data becomes impractical. Databases facilitate the analysis of data generated by Snort sensors. The rules utilized by Snort are stored in text files that can be modified using any text editor. Rules are categorized into groups, with each category's rules contained in their respective files. The primary configuration file, snort.conf, references these files. Moreover, alerts are stored in databases or log files for later access by security professionals. As the number of rules increases, the demand for substantial database capacity for Snort also rises. Suricata requires an average amount of data storage. Zeek employs predefined rules, which lessens the storage needed for files.
Load balancing for architectural metrics can be assessed through various factors related to scalability.
A Low Score (+) indicates no scalability for load balancing,
An Average Score (++) suggests limited scalability.
A High Score (+++) reflects a strong capability to distribute traffic into distinct and balanced workloads.
If the traffic on a network interface connected to a Snort instance exceeds its capacity, it's possible to run additional Snort instances and manage the traffic among them. An adaptive load balancing architecture for Snort is explored in [20], resulting in a high score of +++ for this metric. In scenarios where numerous clients attempt to connect to an access point, Suricata IDS clients employ a sophisticated load-balancing method. These clients perform pre-emptive roaming and load balancing through a beacon element, moving from an overloaded access point to one that can accommodate more clients. In terms of load balancing scalability, Zeek wireless is not as effective as Snort and Suricata.
Scores for the architectural metric Multiple Sensor Support can be evaluated based on the following criteria:
Low Score (+): A minimal number of sensors are supported.
Average Score (++): A moderate number of sensors are supported.
High Score (+++): Extensive support for numerous sensors is provided.
In a business setting, it is common to have multiple locations, which necessitates the deployment of Snort sensors. Snort can be implemented as a distributed IDS in various ways within the organization. One method involves connecting multiple sensors to a single central database that collects all data generated by these sensors. Users can then review and analyze this information via a web browser.
Alternatively, Snort sensors may be configured in such a way that they do not have a direct connection to the database server. Instead, these sensors can log data to local files. Subsequently, these files can be periodically transferred to a central server using methods like SCP. The drawback of this method is that the database's data is not strictly "real-time." The latency is dependent on how often data is uploaded to the central database server via SCP. This setup is illustrated in Figure 1 [7].

International Scientific Journal of Engineering and Management (ISJEM) ISSN: 2583-6129
Volume: 04 Issue: 07 | July – 2025 DOI: 10.55041/ISJEM04702
An International Scholarly || Multidisciplinary || Open Access || Indexing in all major Database & Metadata

Figure 1 : Distributed Snort installation with thehelp of tools like SCP and Barnyard [7]



In this assessment, The Snort receives a +++ score. The foundation of Suricata technology is the Distributed Collaborative Intelligence Architecture (DCIA), which provides the most comprehensive wireless intrusion protection. Utilizing a dedicated network of sensors and client-based agents, the DCIA consistently monitors wireless activities for attacks and violations of policy. The sensors also employ an intelligent channel scanning technique to detect traffic across the RF spectrum. Consequently, Suricata achieves a +++ rating as well. Factors such as stream assembly and reorder influence scores for architectural metrics like these.

Limited Score (+): Unable to locate an attack that has been intentionally fragmented and sent in a disordered manner.

Average Score (++): The likelihood of encountering an assault that has been intentionally fragmented and sent in a non-sequential manner is quite low.

High Score (+++): Extremely proficient at identifying attacks that have been intentionally fragmented and transmitted out of sequence.

The frag3 preprocessor utilized by the open-source IDS Snort facilitates target-oriented analysis. With Frag3, overlapping fragments can be reconstructed using the same technique as the destination system. Users can set up the IDS to implement specific fragmentation reassembly rules for particular hosts or networks. When Snort detects overlapping fragments directed toward one of these hosts, it understands the corresponding reassembly policy to utilize, enabling identical fragment reconstruction by both Snort and the target system. Given its capability to identify attacks that have been intentionally fragmented and transmitted out of order, Snort is awarded a +++ score. Out-of-order attacks can also be executed with Suricata and Zeek. The criteria listed below can be applied to evaluate the architectural metric of State Tracking:

Low Score (+): IOT WSNIDS previously struggled due to an inability to identify storms of random traffic.

Average Score (++): IOT WSNIDS faced confusion with storms of random traffic that were less effective.

High Score (+++): IOT WSNIDS possess significant capabilities to recognize unpredictable traffic surges.

Due to its diverse configuration and command-line options for detecting these random traffic storms detailed in the snort configuration file, Snort achieves a high score for metric state tracking. The commands utilized for this purpose are outlined in Table 3. Suricata and Zeek, both of which can also track state, receive a +++ rating.

Table 3 : Snort configuration commands

Command	Description
Enable_decode_drops	Enables the dropping of bad packets identified by decoder (only applicable in inline mode).
Enable_tcpopt_experimental_drops	Enables the dropping of bad packets with experimental TCP option. (only applicable in inline mode).
Enable_tcpopt_obsolete_Drops	Enables the dropping of bad packets with obsolete TCP option. (only applicable in inline mode).
Enable_tcpopt_ttcp_drops	Enables the dropping of bad packets with T/TCP option. (only applicable in inline mode).
Enable_tcpopt_drops	Enables the dropping of bad packets with bad/truncated TCP option (only applicable in inline mode).
Enable_ipopt_drops	Enables the dropping of bad packets with bad/truncated IP options (onlyapplicable in inline mode).

The architectural metric known as Data Pool Selectability can be evaluated based on the following criteria: Low Score (+): inability to determine which data source should be utilized for intrusion analysis. Average Score (++): moderate ability to discern the appropriate data source for intrusion analysis. High Score (+++): strong capability to identify the correct data source for intrusion analysis. Snort, as a highly sophisticated pattern matcher created to detect patterns indicative of network attack traffic, earns a +++ rating for the Data Pool Selectability metric. Daily, Snort can generate thousands of alerts on any network. To examine intrusion data, Snort employs tools such as ACID, SGUIL, SnortSnarf, Snort_stat.pl, and Swatch. Additionally, Suricata and Zeek are also capable of effectively selecting data pools. System Throughput for architectural metrics can be assessed using the following criteria:

Low Score (+): IOT WSNIDS can handle a lower data input rate efficiently.

Average Score (++): IOT WSNIDS can manage an average data input rate effectively.

High Score (+++): IOT WSNIDS can efficiently process a high data input rate.

To enable Snort to operate with an extremely fast connection, the use of unified logging and a unified log reader such as Barnyard is essential. This allows Snort to deliver alerts in binary format as rapidly as possible while another program handles slower tasks, like writing to a database. Both Suricata and Zeek have received ++ scores for the metric of system throughput, even though they process lower data input rates compared to Snort. Figure 2 illustrates the scores of Snort, Suricata, and Zeek IDS.

VI. CONCLUSION AND FUTURE WORK

An IoT Wireless Sensor Network Intrusion Detection System (WSNIDS) can detect unauthorized activities within a wireless sensor network. The design architecture of an IoT WSNIDS presents a significant challenge due to the rapid evolution of wireless sensor network technologies, which complicates the design process of IoT WSNIDS. This paper proposes an evaluation method based on an architectural metrics scorecard to pinpoint deficiencies and areas for

improvement within an IoT WSNIDS. Once the scorecard is created, the appropriate IoT WSNIDS can be selected based on the system's requirements and the importance assigned to these metrics. This research defines various architectural metrics pertinent to IoT WSNIDS. Furthermore, we introduce a scorecard method to evaluate an IoT WSNIDS by rating distinct architectural metrics. Our assessment technique examines well-known IoT WSNIDS such as Snort, Suricata, and Zeek. There is substantial work yet to be undertaken to identify additional architectural metrics, such as those related to anomaly detection, autonomous learning, Host/OS security, interoperability, packet contents, process security, signature-based detection, visibility, and more. This research outlines key architectural metrics vital for an IoT WSNIDS. As insights are gained from the assessment of an IoT WSNIDS, it will become feasible to establish further architectural metrics and their definitions. Future research will also explore the utilization of the evaluation methodology for other related IoT WSNIDS criteria, including logistical, performance, and various quality metrics.

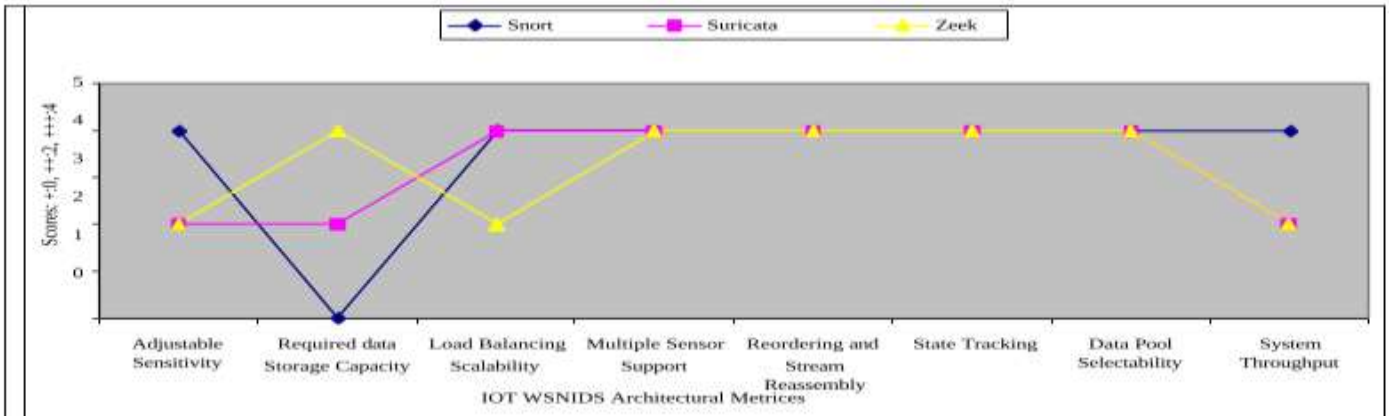


Figure 2: Graph Showing score of Snort, Suricata and Zeek IOT WSNIDS

REFERENCES

1. Rupinder Singh, Dr. Jatinder Singh, "A Metrics-Based Approach to Intrusion Detection System Evaluation for Wireless Network," International Journal of Education and Applied Research (IJEAR)Vol.1, Issue 1, Ver. 1: Jul.-Dec., 2011, ISSN : 2249-4944.

2. H. Asad and I. Gashi, "Dynamical analysis of diversity in rule-based open source network intrusion detection systems," *Empirical Software Engineering*, vol. 27, no. 4, Oct. 2021, doi: 10.1007/s10664-021-10046- w.

3. "Suricata User Guide — Suricata 6.0.4 documentation," *suricata.readthedocs.io*. https://suricata.readthedocs.io/en/suricata- 6.0.4/index.html (accessed Nov. 01, 2021).

4. "suricata-update - A Suricata Rule Update Tool — suricata-update 1.3.0dev0 documentation," *suricata-update.readthedocs.io*. https://suricata-update.readthedocs.io/en/latest/ (accessed Nov. 01, 2021).

5. Roesch, M. (1999). Snort - lightweight intrusion detection for networks. In Proceedings of the 13th USENIX Conference on System Administration (pp. 229-238).

6. http://downloads.visionid.ie/wireless/Dedicated_Distributed_Sensing_The_Right_Approach_to_Wireless_Intrusion_Prevention.pdf

7. Motorola Enterprise WLAN Design Guide Volume November2008Availableat:http://www.scantexas.com/asset/support/Motorola%20Enterprise%20WLN%20Design%20Guide.pdf

8. Stephen Northcutt, "Snort 2.1 Intrusion Detection,"Second Edition, Shroff Publishers, ISBN: 81-7336- 894-9.

9. Harrykar Freelance, "HARRYKAR'S TECHIES BLOG Snort, IDS, IPS, NSM, hacking and beyond," 31 May 09.

10. Rafeeq Ur Rehman,"Intrusion Detection System with Snort Advanced IDS Techniques Using Snort, Apache, MySQL, PHP, and ACID," Prentice Hall, ISBN 0-13-1407 33-3.Availableat:http://ptgmedia.pearsoncmg.com/images/0131473333/downloads/ 0131407333.pdf

11. G. A. Fink, B. L. Chappell, T. G. Turner, and K. F. O'Donoghue, "A Metrics - Based Approach to Intrusion Detection System Evaluation for Distributed Real – Time Systems," WPDRTS April 2002, Ft. Lauderdale, Florida.

12. http://www.suricata.io

13. Roesch, M., & Porras, P. (2013). Snort: The definitive guide. " O'Reilly Media, Inc.".

14. SNORT Users Manual 2.9.0, Snort Project, March 2011

15. J. Gómez, C. Gil, N. Padilla1, R. Baños, and C. Jiménez, "Design of a Snor – Based Hybrid Intrusion Detection System," IWANN 2009, Part II, LNCS 5518, 2009.

16. Reijo Savola, "On the Feasibility of Utilizing Security Metrics in Software Intensive Systems," IJCSNS, VOL. 10 No.1, January 2010.

17. https://en.wikipedia.org/wiki/suricata

18. Snehal Boob and Priyanka Jadhav, "Wireless Intrusion Detection System," International Journal of Computer Applications (0975–8887) Volume 5– No.8, August 2010.

19. http://www.snort.org/

20. SNORT Users Manual 2.9.1, the Snort Project September 20, 2011. Available at: http://www.snort.org/. assets/166/snort_manual.pdf

21. https://old.zeek.org/manual/2.5.5/broids/index.html

22. http://www.networkcomputing.com/wireless/22962263?printer_friendly=this-page
23. M. Alam, Qasim Javed, M. Akbar, "Adaptive load balancing architecture for snort,". http://www.geocities.ws/raza_nust/incc.pdf
24. Deepshikha Bhargava, B.Prasanalakshmi, Thavavel Vaiyapuri, Hemaid Alsulami, Suhail H. Serbaya, and Abdul Wahab Rahmani. "CUCKOO-ANN Based Novel Energy-Efficient Optimization Technique for IoT Sensor Node Modelling." Wireless Communications and Mobile Computing, Vol. 2022, pp.1-9. https://doi.org/10.1155/2022/8660245