RESEARCH ARTICLE OPEN ACCESS

Evaluating Internet of Things Wireless Sensor Network Intrusion Detection System based An Architectural Metrics Scorecard Based Approach

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

[1] Department of Computer Science, Khalsa College Amritsar

ABSTRACT

The anticipated scope, the architecture of IOT WSNIDS, and how they align with the deployment architecture are all compared using IOT WSNIDS architectural metrics. These metrics can be utilized to evaluate an IOT WSNIDS architectural efficiency and to aid in the design of efficient IOT WSNIDS. IOT WSNIDS play a significant role in the security of wireless sensor networks by analyzing wireless-specific traffic, including scanning for external users trying to link to the network through access points. As wireless technology evolves frequently, designing IOT WSNIDS is a challenging work. Architectural metrics can play a significant part in the design of IOT WSNIDS by assessing the sections that are problematic for the architecture of an IOT WSNIDS. We examine a variety of architectural metrics that are pertinent to IOT WSNIDS in this study. The central focus of testing and assessing an IOT WSNIDS is a "scorecard" containing the collection of values. A IOT WSNIDS can be evaluated by giving different architectural metrics related to IOT WSNIDS a score. We use three well-known IOT WSNIDS, Snort, OSSEC, and Bro, as examples of how to use our architectural metrics scorecard-based evaluation technique. Finally, we discuss the outcomes and profound opportunity for further research in this field.

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

I. INTRODUCTION

IOT WSNIDS has ushered in a brand-new, amazing world. Every day, its technology improves, and its popularity rises. The main issue with IOT WSNIDS, however, has been security. For a while, IOT WSNIDS had very scant security, if any, on a wide- open medium. The IOT WSN Intrusion Detection System is a fresh approach to help solve this issue, along with enhanced encryption techniques. A hardware or software program known as an intrusion detection system (IDS) monitors network and/or system activity for malicious behaviour or policy violations and generates reports for a management station (Wikipedia, 2012). This is only done for the wireless network by a wireless IDS. This technology keeps an eye on network traffic for vulnerabilities and alerts staff to take action.

"If you cannot measure it, you cannot improve it," Lord Kelvin once stated. This fact also holds true for concerns about wireless network security. This well acknowledged management theory applies to security as well; an activity cannot be managed if it cannot be measured. Metrics can be a useful tool for security providers to assess the efficacy of various security programs components. Metrics have a significant impact on IOT WSNIDS design. Since the field of wireless network security is still in its infancy, it is difficult to define security metrics for this technology. There is still a lack of a common vocabulary and best practices that are well-documented [1].

In order to evaluate intrusion detection systems, which are now popular for IOT WSN in the commercial sector, this article offers an architectural metrics scorecard-based methodology. We outline a testing approach we created to assess IOT WSNIDS by giving scores to different architectural metrics that are relevant to it. The methodology used in this study compares IOT WSNIDS against a set of architectural metrics that are relevant to IOT WSNIDS, rather than one another.

Systems with any wireless requirements will be able to customize the evaluation of ID technologies to meet their unique requirements owing to this paper's generalized approach. The evaluation may be expanded to include additional measures such as logistical metrics, performance metrics, quality metrics, etc. since evaluation corresponds to a static set of architectural metrics. This paper's standard comparison strategy also provides us with scientific reproducibility.

II. SNORT, OSSEC AND BRO IDS

We chose three IOT WSNIDS—Snort, OSSEC, and Bro—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

A formidable open-source intrusion detection and prevention system (IDS and IPS), SNORT offers real-time network traffic analysis and data packet tracking. To find potentially malicious activities, SNORT employs a rule-based language that integrates anomaly, protocol, and signature inspection methods.

Network administrators can detect Common Gateway Interface (CGI) assaults, buffer overflows, stealth port scans, and denial-of-service (DoS) and distributed DoS (DDoS) attacks using SNORT. A set of rules developed by SNORT characterize malicious network activity, spot malicious packets, and notify users.

SNORT is a piece of open-source software that is available for personal as well as commercial use. Which network traffic should be gathered and what should happen when malicious packets are detected are determined by the SNORT rule

language. This snorting function can be used to find illicit packets in the same way that sniffers and network intrusion detection systems do, or as a full network IPS solution that keeps an eye on network traffic and finds and prevents potential attack channels.

(b) OSSEC

It purports to be the most commonly utilized open source host-based intrusion detection system in the world. We can refer to it as HIDS in brief. It performs rootkit detection, Windows registry monitoring, logging analysis, integrity checking, time-based alerts, and active reaction. There are two components to this: an Ossec server and an Ossec agent. Other servers, which we refer to as Ossec agents, are monitored by the Ossec server. An agent can be added and withdrawn at any moment from the Ossec server for monitoring purposes. We will talk about how to make connections with the server and agent in order to do that. Additionally, it offers a Web interface for displaying all warnings, logs, and agent data.

(c) Bro

The Bro network analysis platform offers both more comprehensive network traffic analysis and network security monitoring. Based on the characteristics and substance of the traffic, Bro analyses network traffic and looks for intrusion attempts. By comparing network traffic with rules specifying occurrences that are judged problematic, Bro can identify intrusions. These guidelines may list actions (such as specific hosts connecting to specific services), indicate which actions merit an alert (such as tries to a particular number of distinct hosts constituting a "scan"), or include signatures for known attacks or ways to access known vulnerabilities. Bro can be told to write a log entry or start the execution of an operating system command if it notices something unusual. Bro focuses on high-volume, high-speed intrusion detection (Gbps). The performance required to do so while running on commercially accessible PC hardware is achieved by Bro by carefully utilizing packet filtering techniques. As a result, Bro can be used as an inexpensive tool for evaluating a site's Internet connection.

III AN APPROACH BASED ON SCORECARDS FOR ARCHITECTURAL METRICS

(a) Developing Scorecard

A "scorecard" including the collection of architectural criteria and their definitions will serve as the main focus of testing and evaluating IOT WSNIDS. Each metric has a possible score of low (+), moderate (++), or high (+++), with higher scores denoting more substantial ratings.

The architectural metrics employed are broad traits pertinent to the architecture of an IOT WSNIDS. Both analysis (such as source code analysis) and open source content (such as specifications, white papers, or reviews provided by traders or users) can be utilized to determine each architectural metric value. To analyze each architectural metric for IOT WSNIDS, we leverage on open source materials. We explore published conference materials (proceedings), research papers, reports, product manuals, and other materials that are open for public scrutiny.

(b) Architectural Metrics for a IOT WSNIDS

The anticipated scope and design of IOT WSN IDS are compared to the deployment architecture using architectural metrics. These metrices assess an IOT WSNIDS's architectural effectiveness [15]. Table 1 displays the metrices defined in that area. Anomaly Based, Autonomous Learning, Host/OS Security, Interoperability, Package Contents, Process Security, Signature Based, and Visibility are other architectural metrics that could be exploited [7].

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.

(c)Architectural Metrics Scorecard Based Approach

We will apply the aforementioned method to the well-known IOT WSNIDSs Snort, OSSEC, and Bro in this section. We decided to evaluate these three because they are among the most popular and operate in different manners. The scoring system for architectural metrics connected to these three IOT WSNIDS is described below with reference to Table 2.

Scores for the architectural metric Adjustable Sensitivity may be determined by the following factors:

Low Score (+): No Adjustability.

Average Score (++): Adjustability via static methods.

High Score (+++): Intelligent, dynamic Adjustability.

Table 2 : Scorecard for Snort, OSSEC, and Bro IDS. + : Low score ; ++ : Average score ; +++ : High Score

Architectural Metrics	Snort	OSSEC	Bro
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), which is a component of Snort, decodes SSL and TLS traffic and, if necessary, determines whether and when Snort should stop inspecting it. In order to improve efficiency and decrease the likelihood of false positive and false negative errors, Snort ignores encrypted traffic [17]. In light of this, Snort receives a high scoring (+++) for metric adjustable sensitivity. Based on fingerprints, Bro sends notifications (for particular nets tumbler versions). Nets tumbler sends out distinctive packets in an effort to reveal a network's SSID. Although it is not always done, when it is, the likelihood of false positives is extremely low. Therefore, Bro receives the average score metric adjustable sensitivity. OSSEC produced a false positive for a Nets tumbler scan that was actually one of the test laptops pinging an AP, as stated in. The Nets tumbler signature needs some work, according to OSSEC. It so receives a score of average for metric adjustable sensitivity.

Architectural metric Required Data Storage Capacity can be assigned score depending on the following criteria:

Low Score (+): Log and other files need to be stored in large capacity storage.

Average Score (++): Storage of medium capacity is required for the log and other files.

High Score (+++): Log and other files must be stored on low capacity storage.

Snort uses databases to store log and alerting data. For smaller applications, logging data to files on the disc is acceptable. However, when there are multiple Snort sensors or the need to keep historical data as well, keeping log data in disc files is inappropriate. Databases also make it possible to analyze data produced by Snort sensors. The rules used by Snort are kept in text files that can be edited with a text editor. Categories are used to group rules. Each category's rules are kept in their own files. The main configuration file snort.conf then contains these files. Additionally, alerts are kept in databases or log files so that security professionals can access them later. As its rules expand and this measure rises, Snort requires a huge database. OSSEC uses average data storage. Bro uses predefine rules, which reduces the amount of storage needed to store files.

Load balancing for architectural metrics Scores for scalability can be determined by the following factors:

Low Score (+): No scalability for load balancing.

Average Score (++): low scalability for load balancing.

High Score (+++): Highly effective at dividing traffic into distinctive, balanced sensor loads.

It is possible to run additional Snort instances and load balance the traffic among them if the network interface connected to a Snort instance is passing more traffic than it can handle. An adaptive load balancing architecture for snort is discussed in [20]. Snort thus receives a +++ score on this metric.

When too many clients try to connect to an access point, OSSEC IDS clients use an intricate load-balancing technique. The clients do pre-emptive roaming and load balancing using a beacon element, switching from an AP that is overloaded to one that is under loaded. When it comes to load balancing scalability, Bro wireless is less effective than Snort and OSSEC.

Scores for the architectural metric Multiple Sensor Support can be determined by the following factors:

Low Score (+): Very few sensors are supported.

Average Score (++): Average amount of sensors supported.

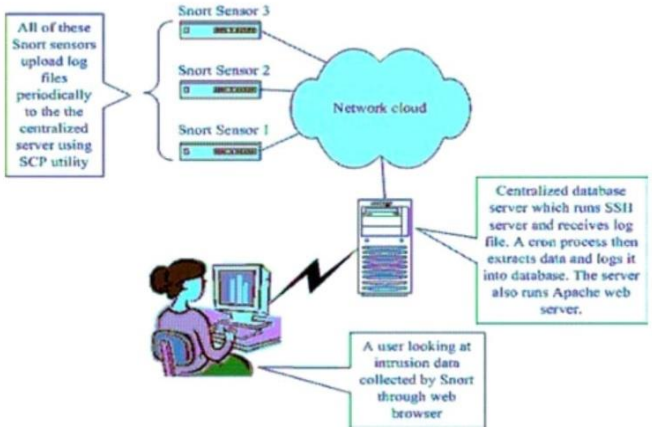
High Score (+++): Support for a large number of sensors.

Multiple locations are likely present in a business environment, necessitating the installation of Snort sensors. Snort can be set up and installed in the business as a distributed IDS in a variety of ways. Connecting several sensors to a single central database is one approach. The database contains all of the data produced by these sensors. The user can then examine and analyse this data using a web browser.

Snort sensors use an alternative approach in which they are not directly connected to the database server. The sensors can be set up to log information to local files. Afterward, these

files can be routinely uploaded to a central server through like SCP. The database's data is not strictly "real-time," which is the sole issue with this method. Depending on how frequently data is uploaded using SCP to the central database server, there is a certain latency. This arrangement is displayed in Figure 1 [7].

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



For this metric, The Snort obtains a +++ scoring. The Distributed Collaborative Intelligence Architecture (DCIA) is the foundation of the OSSEC technology, which offers the most thorough wireless intrusion protect. Using a specialized network of sensors and embedded client-based agents, DCIA continuously scans the wireless activity for assaults and policy breaches. The sensors also make use of a smart channel scanning method to find the traffic across RF spectrum. OSSEC so obtains a +++ rating as well.

Scores for architectural metrics like as stream assembly and reorder can be determined by the following factors:

Low Score (+): No capacity to track down an attack that was deliberately fragmented up and transmitting out of order.

Average Score (++): Finding an assault that has been deliberately fragmented and transmitted out of sequence is extremely less.

High Score (+++): Highly effective at detecting attacks that have been deliberately fragmented and transmitted out of order.

The frag3 preprocessor is used by the open source IDS Snort to implement target-based analysis. With Frag3, overlapping fragments can be reassembled using the same method as the destination host. A user can configure the IDS to apply certain fragmentation reassembly policies for specific hosts or networks. When Snort discovers overlapping fragments bound for one of these hosts, it knows the appropriate reassembly policy to carry out, allowing identical fragment reassembly by Snort and the target host.

Due to its ability to detect attacks that have been deliberately fragmented and transmitted out of order, Snort receives a +++ scoring. Out of order attacks are also possible with OSSEC and Bro.

The following criteria may be used to score the architectural metric State Tracking:

Low Score (+): IOT WSNIDS used to be confused by the lack of a capability to detect storms of random traffic.

Average Score (++): IOT WSNIDS used to be confused by storms of random traffic that are less capable.

High Score (+++): IOT WSNIDS have ample ability to identify random traffic storms. Since Snort has a variety of configuration and command-line options to detect random traffic storms that can be stipulated in the snort configuration file, Snort receives a high score for metric state tracking. Such commands are described in Table 3. Also capable of tracking state, OSSEC and Bro get a +++ scoring.

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_e xperimental_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).

Depending on the following factors, the architectural metric Data Pool Selectability can be scored:

Low Score (+): incapacity to figure out the data source that to be used for the intrusion analysis.

Average Score (++): Average ability to identify the data source that to be used for intrusion analysis.

High Score (+++): High capability to identify the data source that to be used for intrusion analysis.

As a highly complex pattern matcher designed to find patterns of network attack traffic, Snort get a +++ scoring for metric data pool selectability. Every day, Snort can send out thousands of signals on any given network. To analyse intrusion data, Snort uses the tools ACID, SGUIL, SnortSnarf, Snort_stat.pl, and Swatch. It is also possible for OSSEC and Bro to choose data pools effectively.

System Throughput for architectural metrics can be scored using the following factors:

Low Score (+): IOT WSNIDS can

effectively process less data input rate.

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

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

Use of unified logging and a unified log reader like barnyard

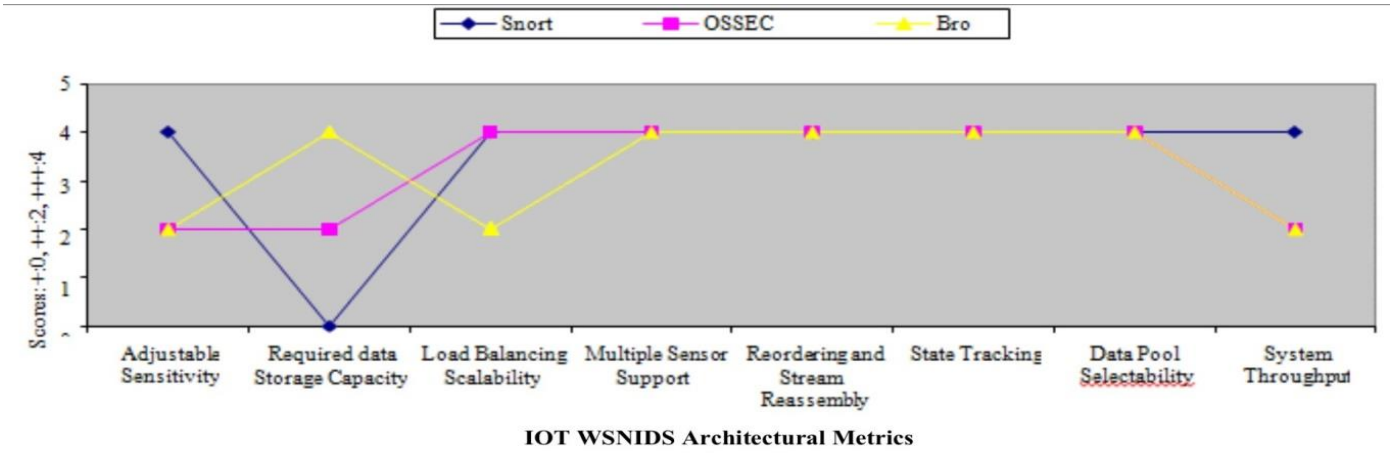


Figure 2 : Graph showing score of Snort , OSSEC and Bro IOT WSNIDS

is required for Snort to function with an extremely swift connection. As a result, Snort is able to report alerts as quickly as possible in binary while another program performs the slow tasks, like writing to a database. Both OSSEC and Bro receive ++ scores for the metric system throughput despite processing less data input rate than snort. Figure 2 displays the Snort, OSSEC, and Bro IDS scores.

IV. CONCLUSION AND FUTURE WORK

A IOT WSNIDS can find unwanted activity on a wireless sensor network. The architectural design of an IOT WSNIDS is a challenging Work since wireless sensor network design technology is evolving quickly, adding to the difficulties in IOT WSNIDS design. In order to identify the areas where an IOT WSNIDS is deficient and requires improvement, this article offers an architectural metrics scorecard-based evaluation method. After generating the scorecard, the suitable IOT WSNIDS may be chosen depending on the necessities of the system and the priority assigned to these metrics.

In this study, several architectural metrics related to IOT WSNIDS are defined. We also present a scorecard technique to assess an IOT WSNIDS by scoring various architectural metrics. We evaluate popular IOT WSNIDS like Snort, OSSEC, and Bro using our evaluation technique. A lot of work needs to be done to identify other architectural metrics, such as anomaly-based, autonomous learning, Host/OS security, interoperability, package contents, process security, signature-based, visibility, etc. This study defines prevalent architectural metrics important to an IOT WSNIDS. As lessons are learned while assessing an IOT WSNIDS, it will be possible to define more architectural metrics and their definitions. Future study will also involve the application of the evaluation methodology to other IOT WSNIDS-related criteria, such as logistical, performance, and other metrics for quality.

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. Roesch, M. (1999). Snort - lightweight intrusion detection for networks. In Proceedings of the 13th USENIX Conference on System Administration (pp. 229-238).
3. http://downloads.visionid.ie/wireless/Dedicated_Distributed_Sensing_The_Right_Approach_to_Wireless_Intrusion_Prevention.pdf
4. Motorola Enterprise WLAN Design Guide Volume November 2008, Availableat:http://www.scantexas.com/asset/support/Motorola%20Enterprise%20WLN%20Design %20Guide.pdf
5. Stephen Northcutt, "Snort 2.1 Intrusion Detection," Second Edition, Shroff Publishers, ISBN: 81-7336- 894-9.
6. Harrykar Freelance, "HARRYKAR'S TECHIES BLOG Snort, IDS, IPS, NSM, hacking and beyond," 31 May 09.
7. 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. Available at: http://ptgmedia. pearsoncmg.com/images/013147333/downloads/ 0131407333.pdf
8. 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.
9. http://www.ossec.net/
10. Roesch, M., & Porras, P. (2013). Snort: The definitive guide. " O'Reilly Media, Inc.".
11. SNORT Users Manual 2.9.0, Snort Project, March 2011
12. 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.
13. Reijo Savola, "On the Feasibility of Utilizing Security Metrics in Software Intensive Systems," IJCSNS, VOL. 10 No.1, January 2010.
14. https://en.wikipedia.org/wiki/OSSEC

15. Snehal Boob and Priyanka Jadhav, "Wireless Intrusion Detection System," International Journal of Computer Applications (0975–8887) Volume 5– No.8, August 2010.
16. http://www.snort.org/
17. SNORT Users Manual 2.9.1, the Snort Project September 20, 2011. Available at: http://www.snort.org/. assets/166/snort_manual.pdf
18. https://old.zeek.org/manual/2.5.5/broids/index.html
19. http://www.networkcomputing.com/wireless/22962263?printer_friendly=this-page
20. M. Alam, Qasim Javed, M. Akbar, "Adaptive load balancing architecture for snort,". http://www.geocities.ws/raza_nust/incc.pdf

[21] 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