International Journal of Research in Engineering, Science and Management
Volume 5, Issue 6, June 2022
https://www.ijresm.com | ISSN (Online): 2581-5792

Implementation of Internet of Things (IoT) Testbed with Distributed Denial of Services (DDoS) Attack Using Cyber Security

B. N. Divya[1], G. R. Neha[2*], S. Nisha[3], R. Pooja[4], R. Tanushree[5]
[1]*Assistant Professor, Department of Electronics and communication Engineering, East West Institute of Technology, Bangalore, India*
[2,3,4,5]*Student, Department of Electronics and communication Engineering, East West Institute of Technology, Bangalore, India*

***Abstract*: Network security and data security are the greatest worries these days. Each association concludes their future business process in light of the past and everyday transactional data. This data may comprise of consumer's confidential data, which should be kept secure. Networks assume significant parts in current life, and cyber security has turned into an imperative exploration region. An Intrusion Detection System (IDS) which is a significant cyber security procedure, screens the condition of programming and equipment running in the network. Most methods utilized in the present IDS can't manage the dynamic and complex nature of cyber-attacks on computer networks. To tackle the above issues, numerous specialists have focused on developing IDSs that gain by machine learning strategies. Machine learning strategies can consequently find the fundamental distinctions between ordinary data and strange data with high precision. In addition, machine learning techniques have solid generalizability, so they are likewise ready to distinguish malicious attacks.**

***Keywords*: Cyber security, Machine learning, intrusion detection system, malicious attack.**

1. Introduction

To acquaint insight into the day-with day gadgets like TV, CAR, AC, and so on, we really want these gadgets to be internet associated. Wireless advances clear the path for this and a wide range of kinds of detecting gadgets sending/receiving data with each through the internet. Over 25 billion 'Things' are internet-associated at this point. The possibility getting an attack in a network is straightforwardly connected with the size of the network. In this way, IoT gadgets are more defenseless against attacks. Security and Data Privacy issues are the central issue in most IoT gadgets and networks. The IoT threads incorporate IoT-explicit threads and general internet acquired threads. An Intrusion Detection System (IDS) is a network traffic monitoring framework that identifies suspicious activity and conveys cautions when it is found. In spite of the fact that intrusion detection frameworks watch out for networks for suspicious activity, they are prone to false alarms. Therefore, when initially introduce IDS items, they should fine-tune them. It involves appropriately arranging intrusion detection frameworks to recognize genuine network traffic and malicious activity. All IDPS advances share one thing for purpose: they can't give 100 percentage precise detection. False positives happen when gentle activity is misidentified as malicious; false negatives happen when malicious activity isn't recognized. It is difficult to take out every single false positive and negatives; as a rule, bringing down one builds the other. Many organizations like to lessen false negatives to the detriment of higher false positives, bringing about additional malicious events being identified however requiring more investigation assets to recognize false positives from authentic malicious events. The intrusion detection framework is the main part for recognizing digital attacks or malicious activities. The proposed framework will distinguish attacks that are named representing a serious threat to IT sectors and services. Existing intrusion detection studies feature progressions and open difficulties, however they disregard IDS execution proficiency, which is significant because of the restricted assets accessible to these gadgets, like CPU, memory, capacity, data transmission, and battery. The objective is to make a network intrusion detector, which is a machine learning algorithm in view of finite automata rule that can recognize bad connections, otherwise called intrusions or attacks.

2. Literature Survey

A. *Survey Paper 1*

Vipin Das, Vijaya Pathak, S proposed the Network Intrusion Detection System Based on Machine Learning Algorithms Network and system security is of paramount importance in the present data communication environment. Hackers and intruders can create many successful attempts to cause the crash of the networks and web services by unauthorized intrusion. New threats and associated solutions to prevent these threats are emerging together with the secured system evolution. Intrusion Detection Systems (IDS) are one of these solutions. The main function of Intrusion Detection System is to protect the resources from threats. They use Rough Set Theory (RST) and Support Vector Machine (SVM) to detect network intrusions.

B. *Survey Paper 2*

Doyen Sahoo, Chenghao Liu Proposed Malicious URL

*Corresponding author: nehagr2412@gmail.com

detection plays a critical role for many cybersecurity applications, and clearly machine learning approaches are a promising direction. In this article, the authors had conducted a comprehensive and systematic survey on Malicious URL Detection using machine learning techniques. In particular, they had offered a systematic formulation of Malicious URL detection from a machine learning perspective. This proposed system aims to provide a comprehensive survey and a structural understanding of Malicious URL Detection techniques using machine learning.

3. Proposed Methodology

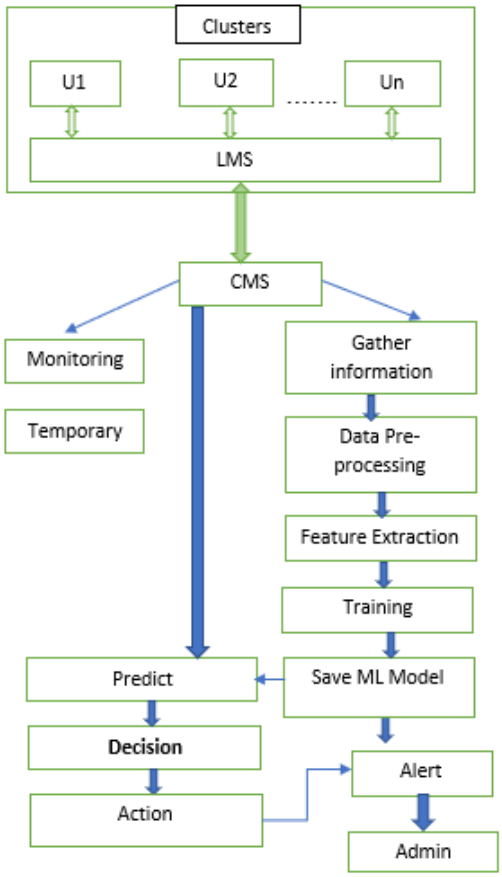

Fig. 1. Block diagram of proposed model

The fundamental idea is to monitor the resources connected in the network. We will isolate the resources into clusters. Each cluster is monitored by a local monitoring device [LMS], the LMS have data of every device has a place with the individual cluster. The access permission of every asset is stacked in the LMS by the centralized monitoring system [CMS] the machine where we have introduced the server. The access permission might resemble file access, applications read write permission, ports utilized, allowed URLs, external device access and so on. The data gathered by the LMS from every asset while the machine is being used are moved to the CMS. The Intrusion Detection System [IDS] works in two folds

First Fold in LMS and another process is running in the CMS. The LMS checks the pattern or rules preloaded. Assuming that any malicious exercises recognized, LMS suspend the asset for quite a while and sends a caution to the CMS. The CMS categorize the threats into three classes like low medium or high threats. The threat classification is classified utilizing the rules defined like recurrence of the exercises or relies upon the sorts of activity. On the off chance that the threat is low threat sends a warning to the asset, on the off chance that it is medium, it blocks the asset for pre-defined time and in the event that it is high, is block the asset and send the caution to the worry department.

The Second Fold depends on the machine learning approach. The threat grouping is marked by the specialists. The machine is trained utilizing the marked data and machine learning based model is stacked into the CMS, so assuming that the thought activity is shipped off the CMS, the trained model will actually want to foresee regardless of whether it is an attack.

Third Fold depends on Smart band device. The smart band device is connected to LMS through the network and it will show the data which has been changed in client system and LMS System and furthermore it will tell the Ip address of the system signed in.

4. System Requirements

A. *Software Requirements*

- Operating System: Windows 10.
- Language: Python, Java,
- Tools: Python IDE, Netbeans IDE, Navicat Premium, MySQL, Apache Tomcat server

B. *Hardware Requirements*

- System: Intel i3
- Memory: 8 GB.
- Hard Disk: 500GB.
- Display board
- ESP8266MCU board
- Battery, Wires

5. Implementation

This model comprises of 4 Stages,

1) System management

In the system management the admin adds the system details connected in the network. Every system detail ought to have system ID, IP address, MAC address, processor details, RAM details, storage details and the cluster ID it has a place with and add Resource Details accessible in the network. The resources might be data storage, printer, and application running in the centralized server.

2) Employee management

In employee management the admin can add the employee details. While adding the employee details admin generate the employee ID, provides employee name, email id, contact number, department and job. This large number of details are put away in the MySQL database present in the server and assign System details admin ought to assign the system that the user will work on. An employee can work on multiple system too i.e., admin can assign multiple systems to a specific employee. But at a time, an employee will actually want to work on a single system. Based on the logged in system, the resources

and access permissions might vary. what's more, we Set Rules, Read/Write Permission has been set by the admin for the users relying upon their department and job. furthermore, the working Hours details will be set by the admin. In a specific system multiple users can work and inside the particular working movement the assigned user will actually want to work on that system. Based on the working movement the resources access might vary. Centralized monitoring system confirms the user's working hours and may impose a few restrictions on the off chance that the user attempts to manipulate something past the working hours and the following is the access Resource Permission When the user signs on the allowed system during the working hours, server sends the asset access permissions and rules into the nearby monitoring device If the LMS distinguishes any violation of any rules or asset access, blocks briefly the user for additional utilization of the system and intimates the centralized monitoring system. The unblocking process can be initiated only by the admin.

3) Monitoring system

In verifying Login Credential When the user signs in to work, needs to provide login credentials. The login credentials are confirmed by the server and if right, user can work and utilize the resources. Number of successful and unsuccessful login attempts. In the event that the login credentials are incorrect, user gets three possibilities. On the off chance that successive three incorrect attempts, centralized monitoring server blocks the user for additional login attempts from any system in the network and intimate the admin. Later admin can check the activities and will actually want to unblock. At the point when the user signs into the system, CMS checks the rundown of the available resources and rules that will be applied on the user and ship off the LMS connected with the cluster. LMS monitors the activities of the user and checks for any violation of the rules and resource access. also, the application running in their system. The rundown of the running application in the user's system can be monitored by the application utilizing task manager. The proposed application, sends the running applications rundown to the LMS. LMS really looks at the permitted applications to be utilized by the user. On the off chance that any violation of the application use, LMS send the alert to the CMS and it thus to the admin. Assuming any external device is utilized LMS block the system immediately and send full alert to the admin. The rundown of the files and registries in the system are put away in the server. Is checked in monitoring system and in the data transmission details the user uploads any data to the server, our application checks the data uploading permission and monitors the network activities. It checks the uploading time and any uploading failures. These activities are logged into the CMS for additional examination. While uploading the data, Once the total data is uploaded to server, CMS generates the hash worth of the got data and contrasts and the got hash. Assuming both the hash values are same there is no progressions of the data in the transmission medium. Assuming that any failures happens while transmission, CMS break down the activity and if over and over happens something similar, finds the jamming issues and the system engaged with it. In the CPU performance LMS continues to monitor the CPU performance of every system inside the cluster in the network performance, LMS monitors the activities occurring in the network. It monitors the uploading and downloading time. LMS estimates the uploading and downloading time based on the data size and the CPU load. Assuming it requires more than the estimated investment sends alert to the CMS for additional examination. Frequency of the activities. The times a specific activity occurring in the network structure a single system is monitored by the LMS inside the predetermined time. Assuming it violates any set rules, LMS blocks the system for that task. Assuming any such activities happen, CMS checks for administration quality and in the event that it degrades the performance, checks for DDoS attack.

4) Machine Learning

In the machine learning stage, we utilize a portion of the models and algorithm i.e., Data Labelling, Data pre-processing, Feature extraction, Train the model and Prediction of the model and the algorithm Random Forest algorithm which gives the highest accuracy for predicting the malware files.

5) Alert

The alert band is connected with the server utilizing ESP 8266. OLED show is fitted with the ESP8266. At the point when any abnormality is detected, server sends the alert to the band and shows on the OLED.

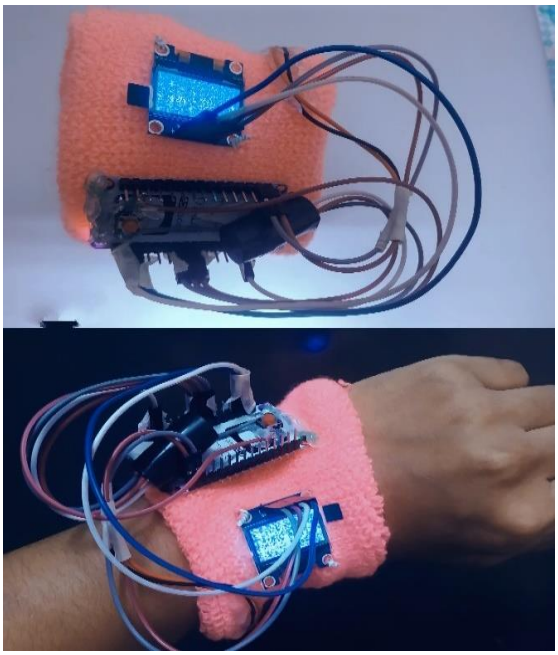

Fig. 2. Proposed model of the band

6. System Outcomes

The centralized monitoring device keeps monitoring the resources in the network in order to detect DDoS attacks. It continuously checks for the login details, resource lists, it ensures accuracy, completeness, consistency and also the validity of organization’s data. It analyses the overall performance.

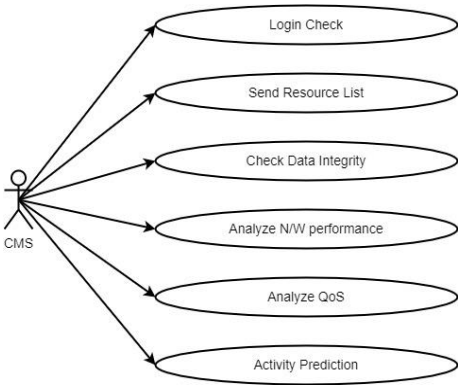

Fig. 3. Activities handled centralized monitoring device

The local monitoring devices keeps monitoring the connected devices and updates to the centralized system. It checks for the activities in the network and also checks for the access of the resources by monitoring the applications, it alerts if there is any external device connected to the device other than the assigned resource. It monitors the file and checks for any irregular activities.

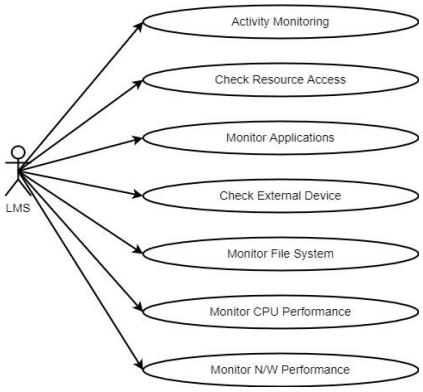

Fig. 4. Activities handled by local monitoring device

Machine learning model will be trained based on the collected data in the network. The trained model will be able to predict the attacks in the network.

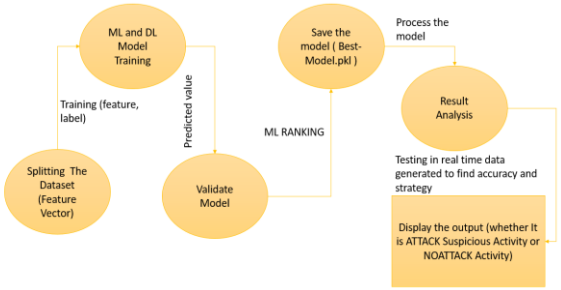

Fig. 5. Flow diagram of machine learning model

Alert: The alert band is connected with the server using ESP 8266. OLED display is fitted with the ESP8266. Whenever any abnormality is detected, sever sends the alert to the band and displays on the OLED.

7. Results

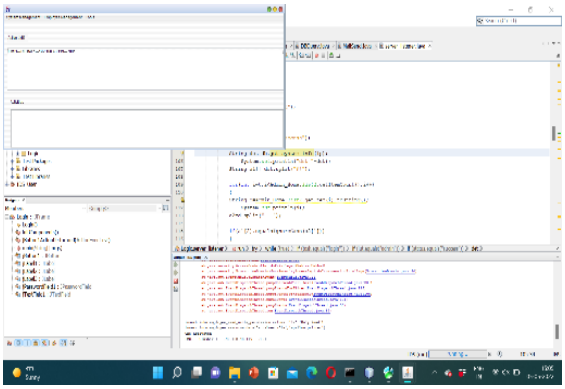

Fig. 6. CMS getting notified by the connected nodes in LMS

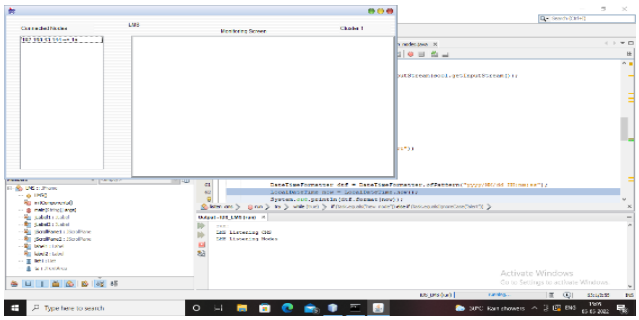

Fig. 7. LMS getting notified about the nodes connected along with Its IP address

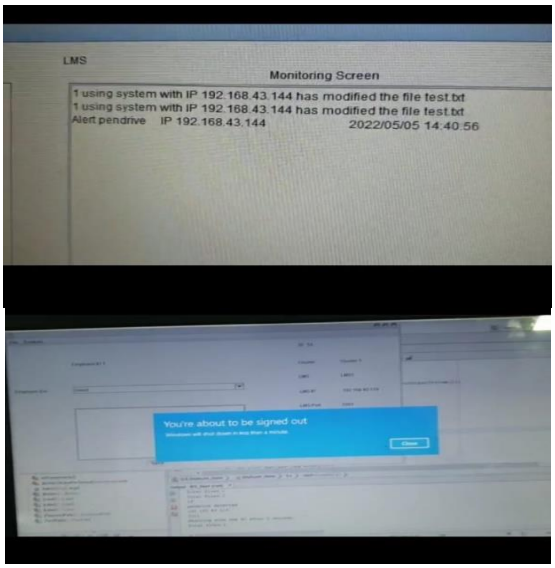

Fig. 8. LMS getting alerted by the external device and user node getting turned off

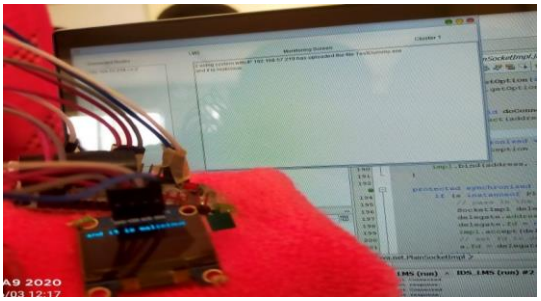

Fig. 9. The alert getting notified in both LMS and Smart band

8. Conclusion

In the proposed system, we have used traditional detection techniques (optimizing pattern) as per static signatures and dynamic detection technology (heuristic). Then, we have chosen for safer system methods as well as speed and modern to rival existing anti-virus. In the future, feature engineering for the available datasets can be done. This feature analysis can be useful to prepare the data for training machine learning based models. The real-time dataset is collected by intruding the DNS amplification attack. Using the collected dataset, a deep learning-based Intrusion Detection System can be framed in the IoT network. The proposal of this work is to find the best solutions to the problems of anti-viruses and improve performance and find possible alternatives for a better working environment without problems with high efficiency and flexibility. We used the optimal traditional methods and modern to detect viruses, for unknown and already detected viruses through the signatures and the Heuristic.

References

[1] Microsoft, "Microsoft security intelligence report", http://www.microsoft.com/technet/security/default.mspx July December 2006.

[2] Dropbox, Inc., dropbox.com webpage, [Online]: https://www.dropbox.com/

[3] C. Grace. "Understanding intrusion-detection systems," [J], PC Network Advisor, vol. 122, pp. 11-15, 2000.

[4] S. Subashini, V. Kavitha S. L., "A survey of security issues in service delivery models of cloud computing," Journal of Network and Computer Applications, pp. 1-11, January 2011.

[5] Shirlei Aparecida de Chaves, Rafael Brundo Uriarte and Carlos Becker Westphall, "Toward an Architecture for Monitoring Private Clouds, December 2011.

[6] Bo Li, Eul Gyu I'm "A signature matching optimization policy for anti-virus programs" 2011.

[7] Chen, Z. & Yoon, J. "IT auditing to assure a secure cloud computing", 2010.

[8] J. Oberheide, E. Cooke, and F. Jahanian "Cloud AV: N-Version Antivirus in the Network Cloud", In Proceedings of the 17th USENIX Security Symposium (Security'08). San Jose, CA, 2008.

[9] Jon Oberheide, Evan Cooke and Farnam Jahanian "Cloud N-Version Antivirus in the Network Cloud", 2007.

[10] Matthias Schmidt, Lars Baumg Artner, Pablo Graubner, David Bock and Bernd Freisleben, "Malware Detection and Kernel Rootkit Prevention in Cloud Computing Environments," 2011.

[11] K. Murad, S. Shirazi, Y. Zikria, and I. Nassar, "Evading Virus Detection Using Code Obfuscation" in Future Generation Information Technology, vol. 6485 of Lecture Notes in Computer Science, pp. 394–401, Springer Berlin, Heidelberg, 2010.

[12] Scott Treadwell, Mian Zhou, "A Heuristic Approach for Detection of Obfuscated Malware", 2009.

[13] Carlin S., & Curran K, "Cloud computing security", International Journal of Ambient Computing and Intelligence.

[14] "Heuristic analysis in Kaspersky Internet Security" [Online]: http://support.kaspersky.com

[15] Algirdas Avizienis, "The n-version approach to fault-tolerant software", IEEE Transactions on Software Engineering, 1985.

[16] Rodrigo Rodrigues, Miguel Castro, and Barbara Liskov. Base, "Using abstraction to improve fault tolerance", in Proceedings of the eighteenth ACM symposium on Operating systems principles, New York, NY, USA, 2001.