NOVATEUR PUBLICATIONS
INTERNATIONAL JOURNAL OF INNOVATIONS IN ENGINEERING RESEARCH AND TECHNOLOGY
[IJIERT] ISSN: 2394-3696 Website: ijiert.org
VOLUME 8, ISSUE 6, June. -2021

REVIEW ON IOT BASED THREATS, VULNERABILITIES & CHALLENGES OF INCIDENT RESPONSE USING EMBEDDED SYSTEM FOR SENSOR NODES

Vaishnavi Rajeshsingh Thakur,
M Tech Student,Dept.of Electronics & Telecommunication Engg.,
CSMSS Chh.Sahu College of Engineering, Aurangabad, India,
Email Id, thakurvaishnavi97@gmail.com

Dr. A. M. Rawate,
Head & Associate Professor,
CSMSS Chh.Sahu College of Engineering, Aurangabad, India,
amr.csmss@gmail.com

Dr. Syeda Sumera Ali,
Associate Professor, CSMSS Chh.Sahu College of Engineering, Aurangabad, India,
syed.sumera.ali@gmail.com

ABSTRACT

The Internet of Things (IoT) is the technology innovation of the century and will forever impact how future generations communicate, work, and handle personal day-to-day tasks. IoT devices streamline processes and often automate everyday household tasks. Despite all the hype and added benefits to their uses, they continue to be the spotlight of recent breaches, privacy concerns, and security vulnerabilities and incidents. The purpose of this study is to shed light on the current threat landscape as it relates to the Internet of Things (IoT) while addressing the reasons why IoT devices are prime targets for attack.Additionally, the study examines the challenges of network defenders, incident responders, and forensic examiners face when investigating incidents.The study found IoT devices are plagued by many software and hardware vulnerabilities, most of which are examined and researched heavily by the Open Web Application Security Project (OWASP). Solutions include providing education to consumers on the risks and mitigations associated some of the more common vulnerabilities. Device manufacturers havea role to play in securing devices before they are released to the general market. Solutions to the challenges faced by Incident Responders and Computer Forensics examiners includes preparation before the incident or crime occurs. Proposed solutions include creating investigative and analytical procedures applying specifically to the Internet of Things. Including understanding their internal makeup, where data is stored, and to whom data is transferred. Answering each of these questions provides potential sources of evidence usedto paint an overall picture of the root cause of incidents and events.

Keywords: Cybersecurity, Professor Michael Sanchez, IoT, Internet of Things, Challenges,Incident Response Process, IoT Forensics

INTRODUCTION

The Internet of Things (IoT) is an emerging paradigm shift in the use of the web from communicating with end-user devices to connecting physical objects by themselves In this paradigm, many objects of daily use around us will be embedded with smart sensorsand computational resources and will be connected to networks in one form or another. Wireless sensor network technologies and modern embedded computing systems will be developed to meet this new emerging paradigm A wireless sensor network (WSN) is a network of sensor nodes that detect and record environmental data and send them to a sink node. The sink node further processes received data and communicates with outer nodes. Thesensor nodes are resource-constrained devices with limited storage and processing capabilities. The Internet Protocol (IP) is a heavyweight protocol that is considered inadequate for the sensor nodes Therefore, a conventional WSN is a non-IP network, where each sensor node only communicates with neighboring nodes or the sink node.

Therefore, a conventional WSN is a non-IP network, where each sensor node only communicates with neighboring nodes or the sink node. However, with the emergence of IoT, the use of IP in resource-constrained WSNs has been requested. In light of this demand, IP version 6 (IPv6) over low-power personal area networks (6LoWPANs) has been standardized. With the advance of 6LoWPAN, it becomes possible to use IP in a resource-constrained sensor node of WSNs. As a result, each sensor node can be accessed anywhere at any time by authorized devices using 6LoWPAN. As an ever- increasing number of emerging WSNs use IP, the number of smart objects connected to the Internet increases. More than 10 billion smart objects will be operated and connected throughthe Internet together with using various applications, such as e-health, gas and electricity meters, etc. The networking company, Cisco, has announced that more than 50 billion devices are expected to be connected to the Internet, producing terabytes of data per second, by 2020. As devices connected to the Internet become increasingly pervasive, security becomes an increasingly critical issue.

Indeed, the IoT devices seamlessly gather personaldata using embedded sensors. Although only authorized users can access these devices, thereis always a certain level of threat that the security system can become vulnerable to illegal activities and attacks, as in desktop computers and server systems. Therefore, an effective security system is needed to protect smart objects against such attacks. An intrusion detectionsystem (IDS) monitors networks for malicious attacks on inbound and outbound packets by using a misuse detection scheme. This scheme uses a pattern-matching algorithm to check predefined pattern sets consisting of intrusion signatures. Each of these signatures is defined in the context of payloads that have previously been revealed as malicious attacks.

The IDS inspects the context of the payloads by checking them against the predefined signature set. Therefore, the pattern-matching engine is one of the most important features innetwork security applications designed to search for malicious patterns. However, it is difficult to operate a conventional pattern-matching engine on smart objects, because most ofthem are resource constrained in terms of power, processing and memory space. In this paper, we propose a new malicious pattern-detecting system that has low computational complexity and requires a small amount of memory in order to protect IoT objects against breaches of security. The proposed system is developed based on a traditional pattern-detection algorithm that is widely used for computer security applications. We have found outthat some of the target data can be skipped without inspecting operations more closely than they are in the traditional security algorithm. The amount of data that can be skipped is precomputed in our proposed system asauxiliary shift values. Furthermore, we limit the memory usage of the traditional algorithm torender it suitable for resource-constrained smart objects. In order to prevent performance degradation due to this limitation, our proposed algorithm reduces the required number of additional matching operations through early decision on character matching operations. We make the following three contributions in this paper to improve the performance of maliciouspattern-matching systems operating on smart objects:

- With the auxiliary shift value, a largeamount of data that are not matched on any patterns can be skipped.
- Within patterns that have identical prefix values, the information obtained by character matching can be used to determine the early termination of the matching operation.
- Proposed algorithm reducesmemory usage for malicious pattern-detecting processes and, therefore, enables the incorporation of resource-constrained smart objects into the security system.

To test the performance of the proposed algorithm, we performed experiments using a smart object embedded with an image sensor and a computing resource. Our algorithm attained a performance gain of 10% over the traditional algorithm by applying auxiliary shifting.

Moreover, the early decision scheme provided an additional speed-up of. Finally, the proposed algorithm provided a maximum speed-up of compared with the traditional algorithm. The rest of the paper is organized as follows. The details of the traditional pattern-matching algorithm and its inefficiency are presented and related research in the area is surveyed. In Section 3, we detail our proposed algorithm. Analyses of computational complexity for the proposed algorithm are presented in Section 4. Experimental results and analyses are presented in This paper is concluded in Section

STATEMENT OF PROBLEM

In the IoT paradigm, physical devices can be connected to the Internet and allowed to utilize a massive amount of data information.

Each device can send and receive necessary data and can make its own decision based on the information. For example, in healthcare applications, the IoT technology can help to cope with emergency situations with rapid responses. Patients will wear medical sensors to monitor their physiological statistics, such as body temperature, blood pressure and breathing activity. Thedata collected from the sensors will be integrated into the global healthcare applications. These applications continuously gather and look over the data in order to monitor the patients. If any kind of unusual conditions are detected, the system alerts the relevant medical center, as well as the patient. Security is one of the most important features to be taken into account in designing the IoT, because of the connectivity and sensitivity of the data collected. Although the IoT networks can only be accessed by authorized users, several kinds of attacks can be mounted against networks. These attacks are aimed at disrupting network communication or collecting private data.

For instance, the Sensors denial-of-service (DOS) attacks that occur in the network layer rapidly disrupt communication. The attacker repeatedly sends packets to the target device and, thus, exhausts its computational resources. An eavesdropping attack is another serious security threat. A malicious node actively steals messages transmitted through the networks and then joins the network as a legitimate node by using the stolen messages. Furthermore, the node carries out attacks against Internet hosts.

An application layer attack causes errors in the operating system of a device and is able tobypass authorized access controls.

SCOPE OF PROJECT

IoT devices threaten the privacy of individuals and the overall securityposture of an organization. Security is often non-existent or requires skills and knowledgeunpossessed by many.

The purpose of this study is to shed light on the current threat landscape as it relates to the Internet of Things (IoT), while addressing the reasons why IoT devices are prime targets for attack. The study explores existing vulnerabilities plaguing theiruse, and the reasons security is not built into the product from their inception.

Additionally, the study examines the challenges of network defenders, incident responders, and forensic examiners face when investigating incidents, and what solutions exist to mitigate the risks associated with the use of IoT devices.

OBJECTIVE OF RESEARCH

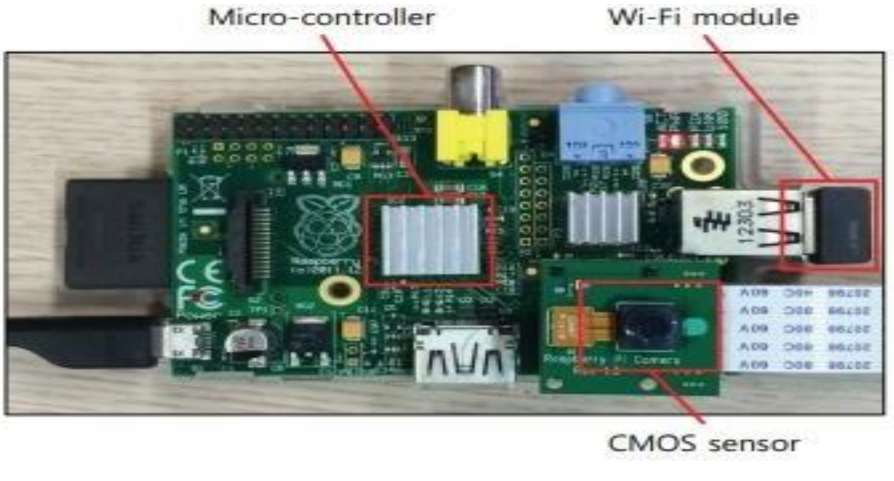
Intrusion detection systems for the IoT networks are required to detect malicious activities in the networks. However, smart objects connected to the IoT networks have lower computing power than the general computing systems, even though theyare more powerful than the traditional sensor node previously used in WSNs.It is difficult toadapt the previously developed security systems for desktop computers to the smart objects due to the hardware restrictions on computing power, memory size and battery life.we will try to implement in high interface time and low power consumption rate Embedding application with hardware support.

Hypothesis and Assumption:

In this section, we evaluate our proposed pattern-matching algorithm as malicious pattern-detection engine for two kinds of security applications.

This engine is implemented on software and tested on a real embedded system.

Secured IoT device Architecture



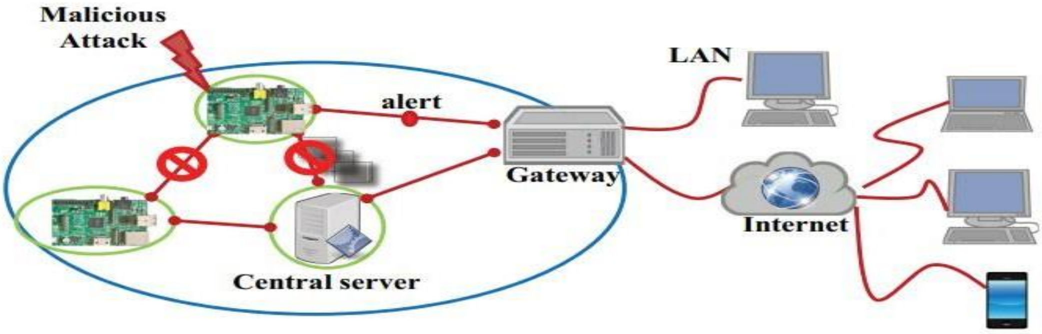


Fig: IoT network consisting of smart physical objects.

Methodology & Tools Used

Malicious Pattern Detection: Most intrusion detection systems detect malicious attacks by using pattern-matching algorithms with predefined malicious attack patterns, as shown in Figure 1. A multiple string matching algorithm has been proposed to find all patterns of a finite pattern set P = {p 1, p 2,….p n}, in a text T = {t0t1 . . . tl−1} of length l.

The patternsand text are sequences of characters from an alphabet Σ. The pattern set is defined according to empirically-determined malicious patterns (malicious signatures). The text is constituted by inbound and outbound packets from networks or system files inside target devices.

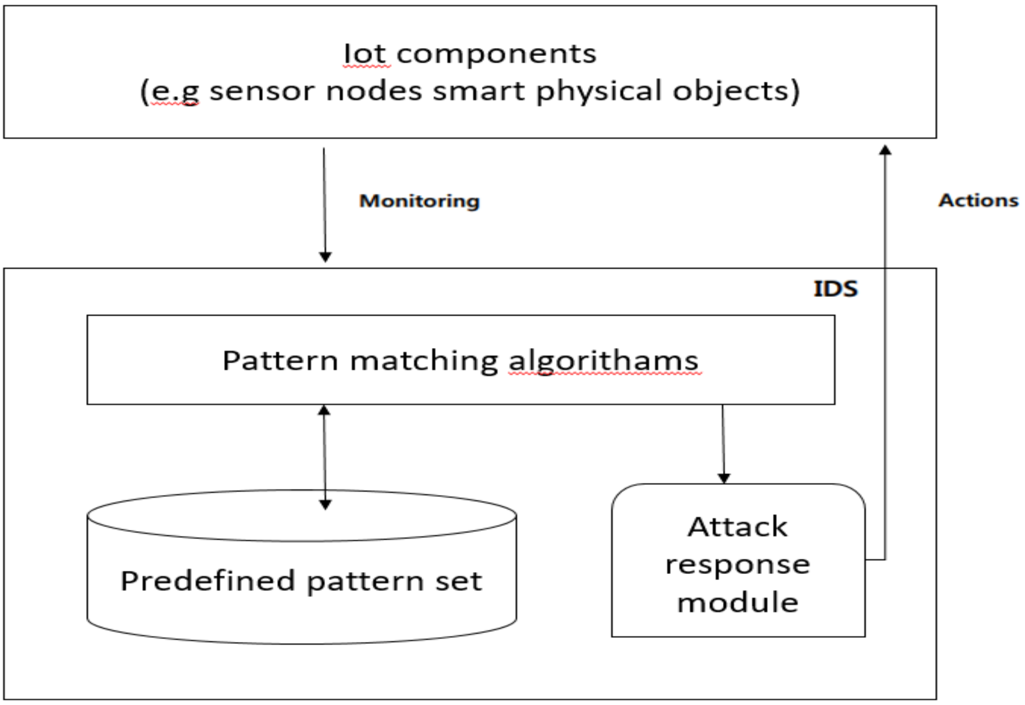


Figure 1. Architecture of intrusion detection system

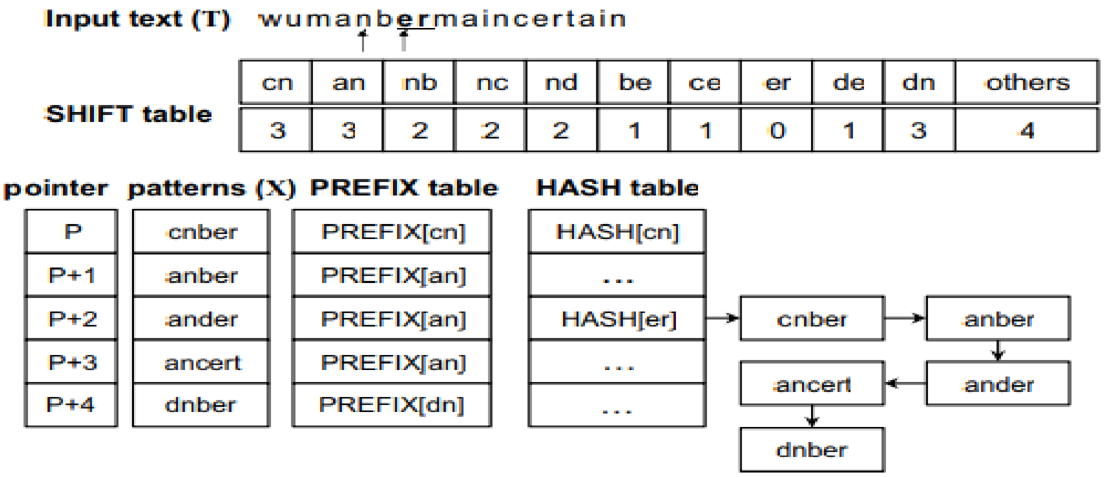
Action

The Wu–Manber (WM) algorithm is one of the fastest multiple pattern-matching algorithms and is widely used as a malicious pattern-detection engine. It is composed of two stages. The first is a preprocessing stage to construct three required tables, the shift, hash, and prefix tables. The second stage is a pattern-matching stage to perform matching operations using these tables. The second stage accounts for most of the execution time. The shift table provides the magnitude of the shift distance for each block. Therefore, the table has Σ B entries to present all possible combinations of characters in the block, where Σ is the number of characters used in the pattern set and B is the size of the block. Each entry of the shift tableindicates the maximum distance to the next possible matching Block The maximum shift distances of the entries can be calculated by a heuristic method using thefollowing two cases:

Case 1: When the block is not related to a pattern's substring consisting of the first m characters of the pattern, the entry has a maximum

value of m−B + 1 (m ≥ B), where m is theminimum pattern length.

Case 2: When the block is related to a substring of the patterns, the entry has the smallest value of m − q, where q is the position with which the block is related in a pattern for B ≤ q ≤ m.
The hash table contains all entries of the shift table with zero shift value. Each entry in thehash table points to the starting address of patterns that have an identical suffix value. The prefix table stores prefix values of the patterns and provides the prefix values before initiatingthe character matching operation.



cn	an	nb	nc	nd	be	ce	er	de	dn	others
3	3	2	2	2	1	1	0	1	3	4

pointer	patterns (X)	PREFIX table	HASH table
P	cnber	PREFIX[cn]	HASH[cn]
P+1	anber	PREFIX[an]	...
P+2	ander	PREFIX[an]	HASH[er]
P+3	ancert	PREFIX[an]	...
P+4	dnber	PREFIX[dn]	...

Figure 2. Wu–Manber multiple pattern-matching algorithm

The matching operation is represented through the example shown in Figure 2, and theoriginal Wu–Manber algorithm is described as follows:

Step 1: Compute the hash value h from the current block and check the shift value of the index h in the shift table. If this shift value is zero, go to the next step. Otherwise, the blockmoves to the right by as many characters as the shift value. Repeat Step 1.

Step 2: Compute the hash value p from the prefix characters of the current block. Match up with the prefix of a pattern listed in the h index in the HASH table. If matched, go to the next step. Otherwise, traverse to the next pattern in the list until all patterns have been checked andthen return to Step 1 after shifting the block by one character.

Step 3: The remaining characters of the pattern, the prefix for which has been matched withp, are compared with the characters of the input text. The character matching operation continues until a mismatch occurs. When a mismatch is found, return to Step 2. When all characters in the pattern match those of the input text, the pattern is found. Return to Step 2 In the figure, h and p are represented as nb and um, respectively (assuming that the size of theblock and that of the prefix is two characters).
Since the index nb in the shift table has the shift value two, the procedure goes to Step 1 after moving the scan window by two characters. Following this, h and p become er and an, respectively, and the shift value of the index er is zero. Thus, the procedure advances to Step 2. The five prefixes of the patterns listed in the er index in the hash table are subsequently evaluated using an, whereas anber, ander and ancert should perform character matching operations on the input text according toStep 3.

Because all characters in anber match with the input text, the pattern is found. The block hence moves forward by one character.

Masquerading attack refers to conducting malicious activities on a computer system by impersonating another user. Such attacks are difficult to detect with standard intrusion detection sensors when they are carried out by insiders who have the knowledge of the system.

One approach to detect masquerading attacks is to build user profiles and monitor for significant changes in user’s behavior at runtime. Intrusion detectorsbased on this principle typically have used user command line data to build such profiles.

This data does not represent user’s complete behavior in a graphical user interface (GUI)- based system and hence is not sufficient to quickly and accurately detect masquerade attacks.

In this chapter, we present a new empirically driven framework for creating a unique feature set for user behavior monitoring on GUI-based systems. For proof-of-concept demonstration,we use a small set of real user behavior data from live systems and extract parameters to construct these feature vectors. The feature vectors contain user information such as mouse speed, distance, angles, and amount of clicks, and keystroke dynamics during a user session.

Formulate our technique of user identification and masquerade detection as a binary classification problem and use Support Vector Machine (SVM) to learn and classify user actions as intrusive or benign.

Proposed technique based on these feature vectors canprovide detection rates of up to 96% with low false positive rates. We have tested our technique with various feature vector parameters and concluded that these feature vectors canprovide unique and comprehensive user behavior information and are powerful enough to detect masqueraders.

CONCLUSION

In this paper, we proposed a novel multiple pattern-matching algorithm for embedded security systems. Since the general embedded systems have a small size for the main memory, we limited the memory usage of the pattern-matching process. However, this limitation leads to performance degradation. We will try to implement in high interface time and low power consumption rate Embedding application with hardware support.

- To reduce the workload of the process, we proposed the auxiliary shifting method and the early decision scheme.
- The proposed methodssuccessfully reduce the workload by skipping a large number of unnecessary matching operations through auxiliary shift values.
- Matching operations, according to the prefix and character values, reduces the complexity of the prefix and character matching operations to a logarithmic scale. Experiments showed thatour proposed method achieved a speedup of up to 2.14 compared to the traditional pattern- matching algorithm given restricted resources.
- The proposed algorithm showed enhanced performance results especially when the number of patterns became large. Our proposed algorithm can thus contribute a high level of scalability to prevalent multiple pattern- matching algorithms.

REFERENCES

1) Atzori, L.; Iera, A.; Morabito, G. The Internet of Things: A survey. Comput.Netw. 2010, 54, 2787–2805.
2) Gubbi, J.; Buyya, R.; Marusic, S.; Palaniswami, M. Internetof Things (IoT): A vision, architectural elements, and future directions. Future Gener.Comput. Syst. 2013, 29, 1645.