© 2016 IJSRST | Volume 2 | Issue 6 | Print ISSN: 2395-6011 | Online ISSN: 2395-602X
Themed Section: Science and Technology

The Prominent Key Role of Denial - of - Service in Intrusion Detection System for Internet of Things

G. Nanda Kishor Kumar[1], Dr. R. K. Sharma[2]
[1]Research Scholar, Department of CSE Sunrise University, Rajasthan, India
[2]Supervisor, Department of CSE, Sunrise University, Rajasthan, India

ABSTRACT

There are currently more objects connected to the Internet than people in the world. This gap will continue to grow, as more objects gain the ability to directly interface with the Internet. Providing security in IoT is challenging as the devices are resource constrained, the communication links are loss, and the devices use a set of novel IoT technologies such as RPL and 6LoWPAN. Due to this it is easy to attack in IoT network. The proposed system is a novel intrusion detection system for the IoT, which is capable of detecting DoS attack and attacker. The proposed methods use the location information of node and neighbour information to identify the attack and received signal strength to identify attacker nodes. Design of such system will help in securing the IoT network and may prevents such attacks. This method is very energy efficient and only takes fixed number of UDP packets for attack detection; hence it is beneficial for resource constrained environment.

Keywords: Internet of things, Intrusion Detection System, 6LoWPAN (IPv6 over low power personal area network), Denial of Service attack.

I. INTRODUCTION

Primarily, an IDS is concerned with the detection of hostile actions. This network security tool uses either of two main techniques (described in more detail below). The first one, anomaly detection, explores issues in intrusion detection associated with deviations from normal system or user behaviour. The second employs signature detection to discriminate between anomaly or attack patterns (signatures) and known intrusion detection signatures. Both methods have their distinct advantages and disadvantages as well as suitable application areas of intrusion detection.

A denial-of-service (DoS) is any type of attack where the attackers (hackers) attempt to prevent legitimate users from accessing the service. In a DoS attack, the attacker usually sends excessive messages asking the network or server to authenticate requests that have invalid return addresses. The network or server will not be able to find the return address of the attacker when sending the authentication approval, causing the server to wait before closing the connection. When the server closes the connection, the attacker sends more authentication messages with invalid return addresses. Hence, the process of authentication and server wait will begin again, keeping the network or server busy.

The Internet of things (IoT) is the inter-networking of physical devices, vehicles (also referred to as "connected devices" and "smart devices"), buildings, and other items embedded with electronics, software, sensors, actuators, and network connectivity which enable these objects to collect and exchange data.

Internet of Things (IoT) is a fast-growing innovation that will greatly change the way humans live. It can be thought of as the next big step in Internet technology. The changing operating environment

associated with the Internet of Things represents considerable impact to the attack surface and threat environment of the Internet and Internet connected systems. IoT is heterogeneous system consisting of various types of sensors nodes or devices with different kind of technology at each layer. However, due to the limited address space of IPv4, an object in the IoT uses IPv6 to accommodate space in Internet. Objects in the IoT can be devices with sensory capabilities, smart metering, health care sensor etc.[1]

As wireless devices become increasingly pervasive and essential in our daily life, security becomes a critical issue. These inchoate devices and technologies are prone to more threats in future, if not governed adequately. 6LoWPANbased IoT has inherited deficiencies: limited resources in terms of power, processing, memory, space and unreliable communication with respect to packet loss rate, collisions.[2] An adversary can take advantage of these weaknesses to initiate different kinds of attacks. More specifically, denial-of-service (DoS) attacks are considered to have adverse effects in disrupting WSNs' communication; still, effective security mechanisms against DoS attacks are yet to be addressed. This paper studies the vulnerabilities present in IP-based WSNs with a major focus on DoS attacks and analyses the existing solutions and countermeasures. Finally, it presents novel security architecture for detecting DoS attacks in 6LoWPAN-based IoT. The proposed solution is actually integrated within the platform being developed in the ebbits project [3]. Such project aims to semantically integrate the IoT into mainstream enterprise systems and support interoperable, online end-to-end business applications. [4] In fact, the networking features exposed by the ebbits platform are opportunistically exploited to improve the performance of the proposed detection solution.

DoS attack

A DoS attack can be done in a several ways.

The basic types of DoS attack include:

1. Flooding the network to prevent legitimate network traffic
2. Disrupting the connections between two machines, thus preventing access to a service
3. Preventing a particular individual from accessing a service.
4. Disrupting a service to a specific system or individual
5. Disrupting the state of information, such resetting of TCP sessions

Another variant of the DoS is the smurf attack. This involves emails with automatic responses. If someone emails hundreds of email messages with a fake return email address to hundreds of people in an organization with an autoresponder on in their email, the initial sent messages can become thousands sent to the fake email address. If that fake email address actually belongs to someone, this can overwhelm that person's account.

Causes of DoS

DoS attacks can cause the following problems:

1. Ineffective services
2. Inaccessible services
3. Interruption of network traffic
4. Connection interference

II. RELATED WORK

The Internet of Things may be a hot topic in the industry but it's not a new concept. In the early 2000's, Kevin Ashton was laying the groundwork for what would become the Internet of Things (IoT) at MIT's AutoID lab. Ashton was one of the pioneers who conceived this notion as he searched for ways that Proctor & Gamble could improve its business by linking RFID information to the Internet. [5] The concept was simple but powerful. If all objects in daily life were equipped with identifiers and wireless connectivity, these objects could be communicating with each other and be managed by computers. In a 1999 article for the RFID Journal Ashton wrote: "If we

had computers that knew everything there was to know about things—using data they gathered without any help from us we would be able to track and count everything, and greatly reduce waste, loss and cost We need to empower computers with their own means of gathering information, so they can see, hear and smell the world for themselves, in all its random glory. Understand the world without the limitations of human-entered data."1 At the time, this vision required major technology improvements. [6] After all, how would we connect everything on the planet? What type of wireless communications could be built into devices? What changes would need to be made to the existing Internet infrastructure to support billions of new devices communicating? What would power these devices? What must be developed to make the solutions cost effective? There were more questions than answers to the IoT concepts in 1999. Today, many of these obstacles have been solved. The size and cost of wireless radios has dropped tremendously. IPv6 allows us to assign a communications address to billions of devices. There will be billions of objects connecting to the network with the next several years. For example, Cisco's Internet of Things Group (IOTG) predicts there will be over 50 billion connected devices by 2020. [7]

IoT describes a system where items in the physical world, and sensors within or attached to these items, are connected to the Internet via wireless and wired Internet connections. These sensors can use various types of local area connections such as RFID, NFC, Wi-Fi, Bluetooth, and Zigbee. Sensors can also have wide area connectivity such as GSM, GPRS, 3G, and LTE. The Internet of Things will. Typically, [8] IoT is expected to offer advanced connectivity of devices, systems, and services that goes beyond machine-to-machine (M2M) communications and covers a variety of protocols, domains, and applications. The interconnection of these embedded devices (including smart objects), is expected to usher in automation in nearly all fields, while also enabling advanced applications like a smart grid, and expanding to areas such as smart cities.

"Things", [9]in the IoT sense, can refer to a wide variety of devices such as heart monitoring implants, biochip transponders on farm animals, electric clams in coastal waters, automobiles with built-in sensors, DNA analysis devices for environmental/food/pathogen monitoring, or field operation devices that assist fire fighters in search and rescue operations. Legal scholars suggest regarding "Things" as an "inextricable mixture of hardware, software, data and service". [10]

III. METHODOLOGY

To detect DoS attacks in IoT, the detection system itself needs to be immune to DoS attacks. In addition it should be scalable, and applicable to most of the real-world IoT scenarios. These design criteria are considered while developing the DoS detection architecture for IoT. Our DoS detection architecture has been designed to detect DoS attacks in ebbits networks.[11] The DoS detection architecture as reported in Figure 1 represents the 6LoWPAN network integrated with the network manager of ebbits. IDS probe (IDS_P) helps the IDS to listen 6LoWPAN network traffic. The most relevant contributions of this paper are the DoS protection manager and the IDS, which are integrated with the ebbits network manager as the security manager. In the following, firstly ebbits network manager and its components are briefly explained; later the proposed DoS protection manager and its components are explained in detail.

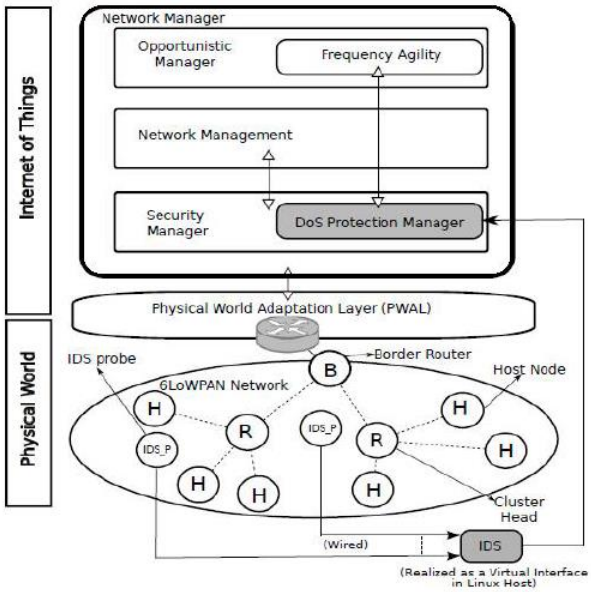

Figure 1. DoS Detection Architecture

IV. DESIGN AND DEVELOPMENT

4.1 System Specification

We run our experiments on Contiki OS, the network simulator Cooja that has shown to produce realistic results. Cooja runs deployable Contiki code. In our simulations, we use emulated Tmote Sky nodes. In general, we expect that the 6BR is not a constrained node and it can be a PC or a laptop; however, currently there exists no PC equivalent 802.15.4 devices, therefore we run the 6BR natively i.e. JNI (Java Native Interface) on Linux. [12] The protocol configuration is as, as Radio interface cc2420 is used, at RDC (Radio Duty Cycling) layer sicslowmac is used, which is 802.15.4 compatible. Above this layer, in MAC CSMA (Carrier Sense Multiple Access) protocol is used. At network layer sicslowpan (6LowPAN), IPv6 and RPL as routing protocol is used. UDP is as transport layer protocol.

4.2 Design DoS attack

A Denial-of-Service attack is an attack which can be used to influence the connection of network, making it inaccessible to its intended users. DoS attack is realized by flooding the target with traffic, or sending it information to triggers a crash [13]. It is one of the most popular cyber-attack methods in security of network. Victims of DoS attack are often the web servers of high-profile organizations such as banking, commerce and media companies The behaviour of each Node must facilitate the propagation and retrieval of valid blocklist Packages throughout the network. [14] While the specifics of Node algorithms are beyond the scope of this paper, certain basic behaviours are vital to running a successful network:

- **Package signature checking**: As described earlier, every Node must use the Publisher's public key to verify digital signatures on all Packages received, dropping any invalid Packages and noting rogue peers.
- **Caching:** Packages moving through Nodes should be cached in local storage to some degree. This provides ample duplication of the blocklist data, allowing several Nodes to answer calls for data.
- **Tracking neighbours**: Nodes must be aware of URLs for other Nodes, perhaps through human collaboration. This knowledge may be shared with other Nodes, provided the neighbours are returning authentic Packages.
- **Package updating**: Packages with newer timestamps (also protected by digital signatures) must invalidate older Packages, and Nodes must make an effort to acquire newer data once a Package has become stale. This should allow fresh data injected by the Publisher to propagate.
- **Content advertising**: Nodes should tell their peers what they have cached locally, in order to help spread the most recent data and facilitate rapid Package lookups in the future.

4.3 Code building blocks

We will be using the following files

- border-router.c
- udp-server.c (udp-client.c can also be used)
- slip-bridge.c (It contains callback function for processing a SLIP connection request)

- httpd-simple.c (A simple web server forwarding page generation to a protothread)

udp-server nodes will form a DAG with the border router set as the root. The border router will receive the prefix through a [SLIP] (Serial Line Interface Protocol) connection and it will be communicated to the rest of the nodes in the RPL network. [5] Refer to the following code snippets in the file border-router.c [15] In this piece of code the node configured as the border router waits for the prefix to be set. Once it receives the prefix, the border router is set as the root of the DAG after which it sets the prefix of the rest of the nodes in the network. Compiling the code The code for RPL border router.

V. SIMULATION RESULTS

The simulation studies involve the deterministic small network topology with 4 nodes and 6 nodes as shown in Fig.2 & 4 respectively. The proposed system is implemented in Contiki OS. We transmitted same size of data packets through various sensors nodes to border router nodes. Proposed system is compared various metrics such as Total Transmission Energy, total number of packets transmitted, network lifetime and energy consumed by each node. We considered the simulation time as a network lifetime and network lifetime is a time when no route is available to transmit the packet.

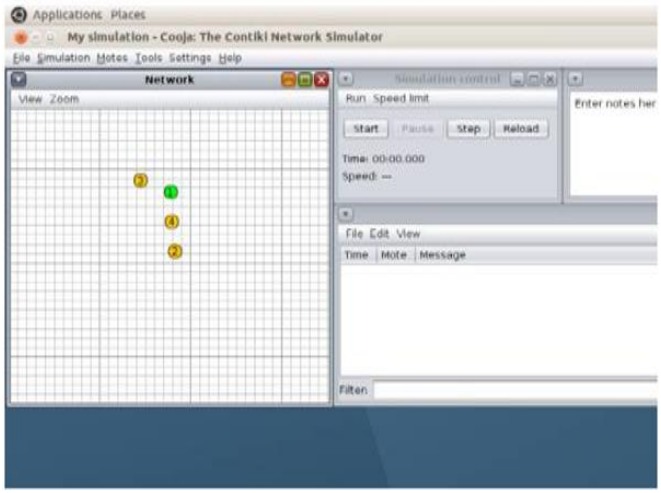

Figure 2. Network window in Cooja tool with 4 nodes

We run our experiments in Contiki's network simulator Cooja that has shown to produce realistic results. Cooja runs deployable Contiki code. In our simulations, we use emulated Tmote Sky nodes. In general, we expect that the 6BR is not a constrained node and it can be a PC or a laptop; however, currently there exists no PC equivalent 802.15.4 devices, therefore we run the 6BR natively i.e. JNI (Java Native Interface) on Linux. The protocol configuration is as, as Radio interface cc2420 is used, at RDC (Radio Duty Cycling) layer sicslowmac is used, which is 802.15.4 compatible. Above this layer, in MAC CSMA (Carrier Sense Multiple Access) protocol is used. At network layer sicslowpan (6LowPAN), IPv6 and RPL as routing protocol is used. UDP is as transport layer protocol.

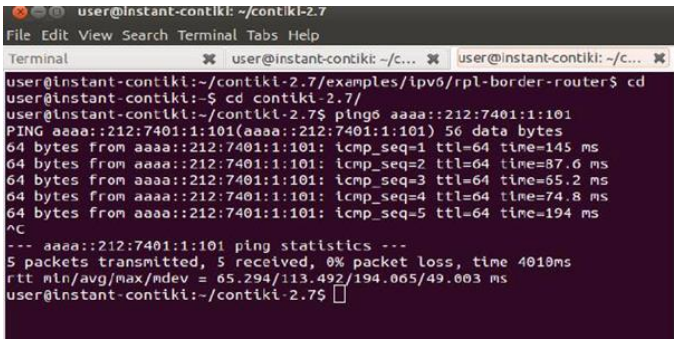

Figure 3. No packet loss during Transmission

In figure 3 shows ping statistics of Border router node 1. As mentioned in the introduction a border router helps in connecting one network to another. In this example the border router is used to route data between an RPL network and an external network. Till now we have only created the RPL network. Now we need to simulate the scenario in which this RPL network is connected to an external network. For this purpose we will use the Tunslip utility provided in Contiki. In this example tunslip creates a bridge between the RPL network and the local machine.

Figure 4. 6 Nodes Topology

```
user@instant-contiki:~/contiki-2.7$ ping6 aaaa::212:7404:4:404
PING aaaa::212:7404:4:404(aaaa::212:7404:4:404) 56 data bytes
64 bytes from aaaa::212:7404:4:404: icmp_seq=1 ttl=62 time=817 ms
64 bytes from aaaa::212:7404:4:404: icmp_seq=2 ttl=62 time=566 ms
64 bytes from aaaa::212:7404:4:404: icmp_seq=3 ttl=62 time=657 ms
64 bytes from aaaa::212:7404:4:404: icmp_seq=4 ttl=62 time=616 ms
64 bytes from aaaa::212:7404:4:404: icmp_seq=5 ttl=62 time=703 ms
64 bytes from aaaa::212:7404:4:404: icmp_seq=6 ttl=62 time=1904 ms
64 bytes from aaaa::212:7404:4:404: icmp_seq=8 ttl=62 time=670 ms
64 bytes from aaaa::212:7404:4:404: icmp_seq=9 ttl=62 time=776 ms
^C
--- aaaa::212:7404:4:404 ping statistics ---
10 packets transmitted, 8 received, 20% packet loss, time 9009ms
rtt min/avg/max/mdev = 566.321/810.088/1904.237/409.264 ms, pipe 2
user@instant-contiki:~/contiki-2.7$
```

Figure 5. Packet loss during Transmission

Figure 4 and 5 shows the nodes topology and Intrusion detection of the system respectively. Due to this the behaviour of Border router may changes. This will see in cooja network simulator.

VI. CONCLUSION AND FUTURE WORK

Considering the potential applications of the IoT it is important that 6LoWPAN networks are protected against internal and external intrusions. This work concludes that, the proposed novel light weight IDS system is basically designed for resource constrained sensor nodes and able to detect Denial of Service (DoS) attacks of two kind packet relay and encapsulation. Mostly centralized modules are used for doing heavy processing and Light weight modules run on sensor nodes causing saving of energy on sensor nodes. Adding location information of nodes made system more efficient for detection of wormhole attack with lesser overhead and with high true positive detection rate. This method takes fixed number of UDP packets for attack detection. The RAM/ROM consumption is also very small as compared to total available sizes. The method give 94% detection rate which is very good for resource constrained environment. In future, we expect to complete the implementation of our proposed architecture and test it against different real attacks. Apart from this, the proposed architecture can be further improved by the following: Distributed Approach; To monitor large networks distributed sniffing, detection mechanisms are required. Security Incident and Event management system (SIEM) once the IDS detect some alerts, this raw information can be accessed by certain alert management software. These tools provide effective statistics and various notifying options to the administrators via email, sms, etc. In future, extending support to SIEMs will be considered. Finally, a centralized monitoring system could be designed such that all network management information from ebbits network manager and IDS alerts could be monitored.

After detecting a DoS attack, specific mechanisms can be designed to defend the attack i.e., the intrusion prevention systems (IPS).

VII. REFERENCES

[1]. Kasinathan, Prabhakaran, et al. "Denial-of-Service detection in 6LoWPAN based internet of things." Wireless and Mobile Computing, Networking and Communications (WiMob), 2013 IEEE 9th International Conference on. IEEE, 2013.

[2]. Pavan Pongle, Gurunath Chavan "Real time Intrusion Detection and Wormhole attack detection in internet of things", by International journal of Computer applications ,2015.

[3]. "Contiki, the open source os for the internet of things." [Online] http://www.contiki-os.org/, Accessed May 2013.

[4]. Raza, Shahid, Linus Wallgren, and Thiemo Voigt. "SVELTE: Real-time intrusion detection in the Internet of Things." Ad hoc networks 11.8 (2013): 2661-2674.

[5]. Tejas mehare , Prof. Mrs. SnehalBhosale " Development of 6LoWPAN BORDER ROUTER for Secure Communication", International Journal of Advanced Research in Computer and Communication Engineering march 2017.

[6]. Jun, Chen, and Chen Chi. "Design of Complex Event- Processing IDS in Internet of Things." Measuring Technology and Mechatronics Automation (ICMTMA), 2014 Sixth International Conference on. IEEE, 2014.

[7]. A. Dunkels, J. Eriksson, N. Finne, N. Tsiftes, Powertrace: NetworkLevel Power Profiling for Low-Power Wireless Networks, 2011.

[8]. Le, Anhtuan, et al. "The impacts of internal threats towards Routing Protocol for Low power and lossy network performance." Computers and Communications (ISCC), 2013 IEEE Symposium on. IEEE, 2013.

[9]. Alessandro Sforzin and Mauro Conti " RPiDS: Raspberry Pi IDS A Fruitful Intrusion Detection System for IoT", IEEE Conferences on Ubiquitous Intelligence & Computing,2016.

[10]. A. Hassanzadeh, Z. Xu, R. Stoleru, G. Gu, and M. Polychronakis, "PRIDE: Practical intrusion detection in resource constrained wireless mesh networks," in Information and Communications Security. Springer, 2013, pp. 213–228.

[11]. A. K. Kyaw, Y. Chen, and J. Joseph, "Pi-IDS: evaluation of opensource intrusion detection systems on Raspberry Pi 2," in 2015 Second International Conference on Information Security and Cyber Forensics (InfoSec). IEEE, 2015, pp. 165–170. 12. D. Evans, "The internet of things: How the next evolution of the internet is changing everything," CISCO white paper, 2011.

[12]. O. Garcia-Morchon, S. Kumar, R. Struik, S. Keoh, and R. Hummen, "Security considerations in the ip-based internet of things." IETF (work in progress) Available [Online] http://tools.ietf.org/html/draft-garciacore-security-05, Mar. 2013.

[13]. H. Sedjelmaci and M. Feham, "Novel, Hybrid Intrusion Detection System for Clustered Wireless Sensor Network," International Journal of Network Security & Its Applications (IJNSA), Vol.3, No.4, July 2011, vol. 3, no. 4, pp. 1–14, 2011.

[14]. Ismail butun , Salvatore D. Morgera and Ravi sankar ,"A Survey of Intrusion Detection Systems in Wireless Sensor networks", IEEE Communications Survey and Tutorials ,2014.

[15]. https://www.techopedia.com/definition/24841/denial-of-service-attack-dos