# Anomaly Detection in IoT with Cooja Simulator

Mochammad Ichsan Rahmat Sanjaya[1], Aji Gautama Putrada[2], Vera Suryani[3]

[1,2,3]Fakultas Informatika, Universitas Telkom, Bandung

[1]ichsanrahmat@students.telkomuniversity.ac.id, [2]ajigps@telkomuniversity.ac.id,
[3]verasuryani@telkomuniversity.ac.id,

Abstract

The idea of Internet of Things (IoT) is implanting networked heterogeneous detector in our daily life. It opens extra channels for information submission and remote control to our physical world. A significant feature of IoT network is that it collects data from network edges. Moreover, human involvement for network and devices maintenance is greatly reduced, which suggest an IoT network to be highly self-manages and self-secured. For the reason that the use of IoT is growing in many important fields, the security issues of IoT need to be properly addressed. Among all, Distributed Denial of Service (DDoS) is one of the most notorious attacking behaviors over network which interrupt and block genuine user requests by flooding the host server with huge number of requests using a group of zombie computers via geographically distributed internet connections. DDoS disrupts service by creating network congestion and disabling normal functions of network components, which is even more destructive for IoT.

Keywords: IoT, DDoS attack defensive mechanism, network communication simulation.

## 1. Introduction

### 1.1 Background

The term of Internet of Things (IoT) is a system of interconnected devices, machines and related software services. IoT plays an important role in the modern society since it enables energy efficient automation for enhancing quality of life. However IoT systems are an obvious target for cyber-attacks because of their ad-hoc and resource-constrained nature. Therefore, continuous monitoring and analysis are needed for securing IoT systems. For the security monitoring and analysis of IoT, forecasting malicious attacks is crucial to adapt with unexpected conditions, take precautions, protect sensitive data, provide continuity and minimize possible losses. Because vast amount of network and sensing data produced by IoT devices and systems.

Anomaly detection in the IoT helps to detect likely errors and possible causes. It is really important for early detection because it can threaten lots of device connected and disable them to do malicious thing.

An IoT DDoS defense for an IoT end network is proposed for preventive measuring and avoiding DDoS attack. In a typical IoT end network involved with DDoS attack scenario, four different types of nodes including working node, monitoring node, legitimate user node, and the attacker node are constructed to be present in a simulation environment. Many researches have proposed DDoS defense technologies over the internet. Others have done work classifying types of DDoS attacks and defense mechanisms. However, not much has been done for addressing and solving DDoS problem specifically over IoT network even though DDoS poses more threats to IoT network because of its open nature.

Bayesian network used to detect anomalies in sensors, because they can handle high dimensional data which humans find difficult to interpret. While some anomalies are clearly visible by plotting often anomalies are far more subtle and based on the interaction of many variables.

### 1.2 Topic and Boundary

In the background of this study, the topic of the problem being solved is how to simulate the attacks on the IoT with Cooja Simulator.

Limitation problems in this study are as follows: the simulation being done in Cooja simulator on Contiki operating systems, with 3 scenarios, the nodes that will simulate the simulation is the working node, attacker node and User node.

### 1.3 Purpose

In designing this system, it is expected to be able to build simulation about the detecting the anomaly of the IoT network using Cooja Simulator and how the attack work and how to defend them.

### 1.4 Writing Organization

Organization of writing in this thesis include: Identification of Problems where the problem will be identified about how the DDoS attack works. Literature Study, looking for references related to DDoS attack as a theoretical basis in providing solutions to problems that occur. Data Collection, collecting data as a data set and test data. Design, describes how the workflow processes the simulation. Analysis of Research Results on whether the output is in accordance with the objectives. Report Writing of the stages that have been carried out as a result of the solution of the problem.

## 2. Related Study

### 2.1 Previous Research

N-BaIoT—Network-Based Detection of IoT Botnet Attacks Using Deep Autoencoders. (2018) [1], due to traffic volume of DDoS attack using botnet detection methods to detect attacks using autoencoders, by the observation tests can be seen that they have a lot of steps and tests to gain results.

Anomaly detection and privacy preservation in Cloud-Centric Internet of Things. (2015)[2],due to system vulnerability the author presents security and privacy risk from anomaly detection aspect, this paper using a lot of anomaly detection methodologies and the approach steps to do is too difficult.

Graph-based anomaly detection. (2003) [3],due to a lot of fraud and intrusion happening the author using graph based detection with involving subdue system, the detection and need to be checked manually whether there is some attacks or not and a detailed examination of the relationship between graph regularity and anomaly detection needed.

DETEKSI ANOMALI MENGGUNAKAN KLASTERISASI GRAF (STUDI KASUS DETEKSI INTRUSI). (2008) [4], due to abuse activity on the internet the author using density based approach and ODIN(Outlier Detection using Indegree Number) algorithm which graph based to detect anomaly in the internet, the result is quite hard to define and needed some algorithm to define the result.

### 2.2 Internet of Things

The Internet of Things (IoT) is a system of interrelated computing devices, mechanical and digital machines, objects, animals or people that are provided with unique identifiers (UIDs) and the ability to transfer data over a network without requiring human-to-human or human-to-computer interaction.

The definition of the Internet of Things has evolved due to the convergence of multiple technologies, real-time analytics, machine learning, commodity sensors, and embedded systems. Traditional fields of embedded systems, wireless sensors networks, control systems, automation (including home and building automation), and others all contribute to enabling the Internet of Things. In the consumer market, IoT technology is most synonymous with products pertaining to the concept of “smart home”, covering devices and appliances (such as lightning fixtures, thermostats, home security systems and cameras, and other home appliances) that support one or more common ecosystem, such as smartphones and smart speaker.

There are a number of serious concerns about dangers in the growth of IoT, especially in the areas of privacy and security; and consequently industry and governmental moves to begin to address these. [5, 6, 7]

## 2.3 DoS Attack

Denial of Service (DoS) is a type of attack on computer or server on the internet network by depleting resources that are owned by the computer cannot perform its function properly so that it indirectly prevents other users from gaining access to services from a computer that attacked. In case of DoS attack, the attacker will try to denied a user access towards system or network with some method such as:

1. Traffic Flooding, to overflow the network traffic so the traffic that come from users cannot get into the network system.

2. Request Flooding is to overflow request towards network service provided by a host so the request from the user cannot served by the provider.

3. Disturbing the communication between host and client that registered with a lot of ways, including to change the system configuration or even destroying a component and server physically.

## 2.4 DDoS Attack

To start with, Denial of Service (DoS) attack is defined as denying and disrupted legitimate access to the service or resources on target server. Even worse, Distributed Denial of Service (DDoS) attack typically engages more computers and internet connections to such attacking behavior to engender real threats that seriously blocks or suspends other users accesses to the host server, which leads to huge business loss and client inconvenience.

The targeted service could be disrupted by the attack crashing the host server with some carefully designed packets whose content causes certain operating system to freeze or reboot. Other than that, the malicious packets occupy all the resources on the host server with massive volumes of bad request, which is also called bandwidth attack in related researches. Prevented by patching the host operating system against the identified attacks, the first form of attack could be stopped at some point. However, the massive volume-based attack is quite hard to defense.

A volume-based attack is usually initiated with installing “bot” onto vulnerable systems. Bot technology was used in industry for automating process. In such way, hackers can easily populate their attacking army with zero cost. Zombies’ or bots’ behavior could be manipulated through secured channels in order to launch further attacks to the targeted IP or a local network.

To specify the difficulties in finding solutions, first, the aggregated large traffic volume exceeds throughput of many network security devices and capacity of corporate internet link. Second, controlled zombie systems are geographically distributed, which is hard to locate source IP addresses. Third, when separately examined, single attack from one source is not powerful enough to be discriminate from a legitimate request, which makes it look similar to a flash crowd created by legitimate requests at a website peak time.[8]

## 2.4 Contiki Operating System

Contiki is an open source operating system for sensor network developed at the Swedish Institute of Computer Science since 2004. Among the available network simulation tools, Contiki operating system holds powerful simulating and communication methodology for the IoT microcontrollers, named ‘motes’ and mentioned as ‘nodes’ in this study. Contiki runs as a virtual machine over an operating system handled by VMware player. So, it is highly portable and efficient for code backing up. To keep the memory overhead down in the resource limited devices, event-driven programming is applied in the operating system. Plus, to ensure the event-driven program easy to write and debug, a thread-like programming style, called protothreads, which helps to reduce the lines of code with only two bytes of memory overhead per protothread. [9, 10]

### 2.5 Cooja Simulator

COOJA is a Contiki network simulator. It stands out from other emulators by allowing cross-level simulation in the WSN. It enables simultaneous simulation from low level regarding that for sensor node hardware to high level regarding that for node behavior. With this simulation environment, developers can see their applications run in large-scale networks also tune the emulated hardware in extreme detail.[11]

### 2.6 Rime Stack

As part of Contiki's system core, rime is a lightweight layered communication stack for sensor networks. It was tailored to simplify the implementation of traditional layered communication protocol in sensor network and encourage code reuse. It fully supports operations like broadcasting, unicasting, network flooding, and address free multi-hop semi-reliable scalable data collection, which makes it a great fundament for building an out-of-tree implantation for the proposed DDoS defending algorithm.[12]

## 3. System Design

### 3.1 System Plan

The system design for this study can be seen in Figure. 1. Based on Figure. 1, the first step that is taken for this research is creating the working node with or without defense mechanism, and then creating attacker node, and simulate the simulation.

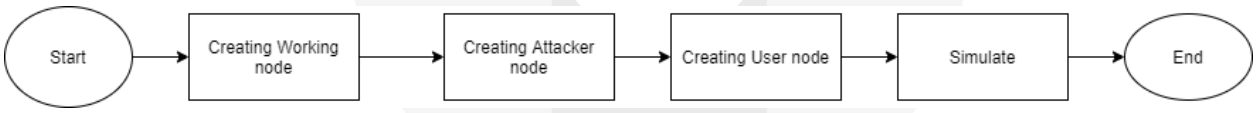


*Figure 1 Design process for the simulation*

### 3.2 Working node

A working node is the device collecting information and executing simple tasks in an IoT network. In one hand, they are characterized by limitation in memory, storage, and power supply. On the other hand, they are usually of the most number in a functioning IoT local network. So, it is necessary to ensure each of them is equipped with certain attack detecting mechanism which also has to be lightweight and inexpensive to implement.

A major behavior of a proposed working node is serving request and defending itself from attacks. During the request serving stage, the service of a node should be blocked by a previously validated request and not be available to server other request when it is busy. The node will notify the requesting entity whether its request has been served. Additionally, the node will not enable queuing function for the rejected requests, which corresponds to the simplicity of the device. As a result, the competition over a limited service is always won by the user who requests most frequently, which in the case of a typical DDoS attack, this role is played by the attacker.

To defend itself from DDoS attack, a node should be able to distinguish malicious requests from legitimate ones. As for the reason that DDoS requests usually contain the same meaningless content, the proposed defending algorithm determines a sender is malicious according to the consistency of the content in the packets it sends. If a sender repeatedly send request with same content, it will be flagged as an attacker. Upon the reception of request from this exact address, the working node will refute its request and remain bandwidth for service providing.

3.3 Attacker node

An attacker's behavior could be differed from that of a legitimate user by its high frequency of sending requests and the same content in those sent packets. To implement this feature in a simulation, an attacking node is designed to always send same request with certain higher frequency compare to that of legitimate user node. To detail, a timer to be expired in random seconds between 1 to 3 second is set, after the initiation of the attacking node, whenever the timer is expired, it broadcast and send same junk packets to the nearby working nodes to ask for service.

3.4 User node

User is distinguished from an attacker by sending request for service with a lower frequency and reasonable content. To implement this feature, a user node is designed to unicast its request with a frequency of 10 seconds after initiation to one of the working nodes in an IoT end network. It will wait and print the response from the working node.

3.5 Simulate

The simulation will take place in Contiki Operating System with Cooja Simulator using Rime stack to communicate each nodes.

## 4.Evaluation

To test the effectiveness of the proposed simulation, several IoT network scenarios were constructed with the four types of proposed nodes. To demonstrate and clarifying the effect of the proposed simulation, interactions between each pair of two different types of nodes are individually tested with and without the defending algorithm.

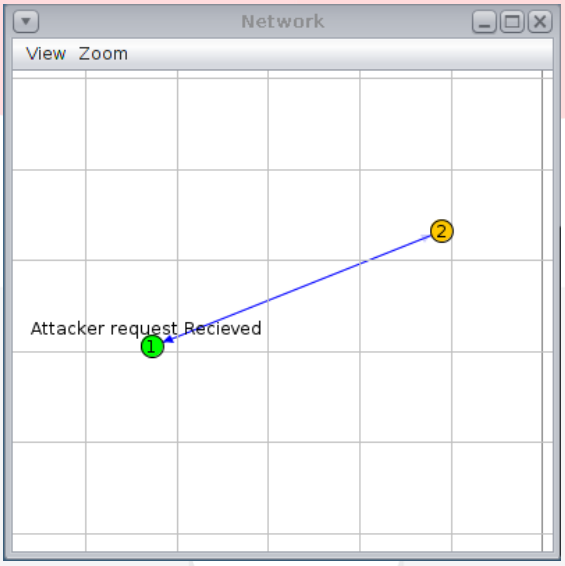


*Figure 2 An attacking node (ID=2) and a working node (ID=1) interacting without defending Algorithm*

| Time(s) | Mote Output | |
|---|---|---|
| | Mote ID | Message |
| 0.517 | 2 | Starting 'Attacker Request' |
| 0.663 | 1 | Starting 'Serve Request' |
| 2.786 | 1 | Request has received 'you are under attack' |
| 4.011 | 2 | Request has served |
| 5.411 | 1 | Request has received 'you are under attack' |
| 7.887 | 2 | Request has served |

*Table 1 Interactive communication flow between an attacking node (ID=2) and a working node (ID=1) without defending algorithm*

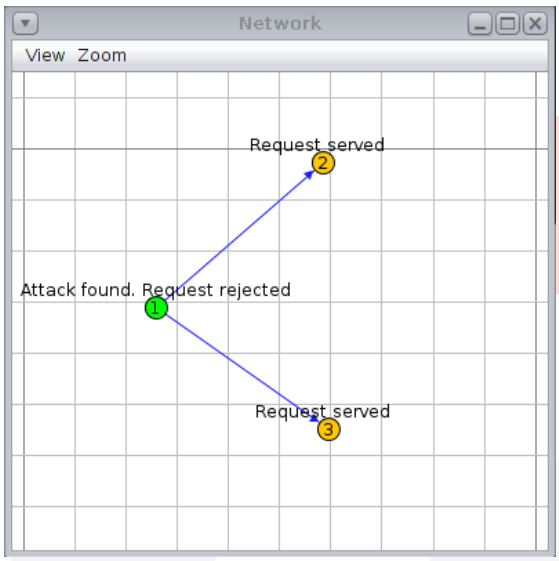


*Figure 3 Two attacking nodes (ID=2, 3) and a working node (ID=1) interacting with defending algorithm.*

| Time (s) | Mote Output | |
|---|---|---|
| | Mote ID | Message |
| 0.517 | 2 | Starting 'Attacker Request' |
| 0.663 | 1 | Starting 'Serve Request' |
| 1.180 | 3 | Starting 'Attacker Request' |
| 2.785 | 1 | Request has received 'you are under attack'<br>Attack found. Request Rejected |
| 4.013 | 2 | Request has served |
| 4.052 | 3 | Request has served |
| 7.659 | 1 | Request has received 'you are under attack'<br>Attack found. Request Rejected |
| 7.801 | 3 | Request has served |
| 7.887 | 2 | Request has served |
| 8.035 | 1 | Request has received 'you are under attack'<br>Attack found. Request Rejected |

*Table 2 Interactive communication flow among a working node (ID=1) and two attacking node (ID=2, 3) with defending algorithm*

**Scenario 1 & 2**

As in the Fig.2 one working node and one attacker node created to simulate, as the result of the output was shown in Table 1 the working node was attacked by attacker node without defending algorithm. And as the Fig.3 one working node and two attacker node created to simultaneously attack the working node with defending algorithm as the result shown at Table 2.

In this scenario, one attacker node and one working node are placed in an IoT local network. The attacker node requests for service every 1 to 2 seconds and will not stop until the end of simulation. The purpose of this scenario is to examine whether the working node could distinguish and reject the malicious service request after it being blocked for the first time. The first set of results (Fig.2, Table 1) shows the situation happened without the defending algorithm. However, with the defending algorithm applied (Fig.3, Table 2), the working node could distinguish the malicious peers and reject their requests after serving them for the first time. The records of malicious nodes are archived in the record list, which is indicated by the growing length in the record list.

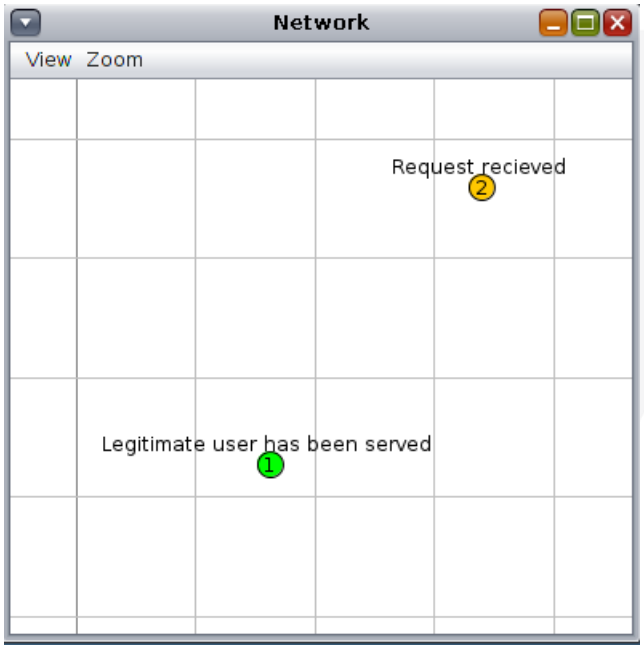


*Figure 4 Interactive communication flow between a working node (ID=1) and a legitimate user node (ID=2) not under attack*

| Time (s) | Mote Output | |
|---|---|---|
| | Mote ID | Message |
| 0.517 | 2 | Starting 'Serve Request' |
| 0.663 | 1 | Starting 'Legitimate user request' |
| 13.959 | 1 | Legitimate user has been served |
| 21.289 | 2 | Request received |
| 27.763 | 1 | Legitimate user has been served |
| 33.898 | 2 | Request received |

*Table 3 . Interactive communication flow between a working node (ID=1) and a legitimate user node (ID=2) not under attack*

**Scenario 3**

As for this scenario shown at Fig.4 one working node and one user node created to communicate with each other without any malicious intent as the result shown at Table 3.

In this scenario, one user node and one working node are placed in an IoT local network (Fig.4, Table 3). The user node starts asking for service after the simulation begins for 10 seconds. The working node is expected to service the request and output the job status. If the request is served, the working node returns the "Served" status with an e num in a unicast message to the user node. Then, the user node will print the message about its request has been served by the node of the responder to indicate the completion. Otherwise, it will send a "Rejected" message back to the user node to notify it being unable to fulfill the request.

**Methodology**

The method that will be used if the simulation was going to be implemented is Bayesian network. Bayesian networks are well suited for anomaly detection because they can handle high dimensional data which humans find difficult to interpret. The prediction(Prognostics) of the Bayesian network to detect the anomalies in advance with some key points such as the inception of failure for when the problem start, the detection point for when the algorithm first deemed the data to be anomalous, and to the actual point of failure for when the system begin to fail. While some anomalies are clearly visible by plotting individual variables, often anomalies are far more subtle, and are based on the interaction of many variables. Bayesian network also support for discrete and continuous variable, also support for high dimensional models, which human are bad at interpreting. [13]

## 5.Conclusion

In the simulation that has been executed, of Anomaly detection in IoT using Cooja Simulator. Based on the results, the proposed simulation could effectively help the working nodes in an IoT network to distinguish malicious requests from attacker ones and process them differently.

For further research the simulation will have more nodes to work with and also defend and detects more malicious attacks towards the IoT.

## References

[1] Meidan, Y., Bohadana, M., Mathov, Y., Mirsky, Y., Shabtai, A., Breitenbacher, D. and Elovici, Y. (2018). N-BaIoT—Network-Based Detection of IoT Botnet Attacks Using Deep Autoencoders. IEEE Pervasive Computing, 17(3), pp.12-22.

[2] Butun, I., Kantarci, B. and Erol-Kantarci, M. (2015). Anomaly detection and privacy preservation in Cloud-Centric Internet of Things. IEEE.

[3] Noble, C. and Cook, D. (2003). Graph-based anomaly detection. Proceedings of the ACM SIGKDD International Conference on Knowledge Discovery and Data Mining.

[4] Akhsanajaya, A. (2008). DETEKSI ANOMALI MENGGUNAKAN KLASTERISASI GRAF (STUDI KASUS DETEKSI INTRUSI). open library Telkom university. [online] Available at: https://openlibrary.telkomuniversity.ac.id/home/catalog/id/94717/slug/deteksianomali-menggunakan-klasterisasi-graf-studi-kasus-deteksi-intrusi-.html [Accessed 20 Nov. 2019].

[5] Zhang, Daqiang, Laurence T. Yang, and Hongyu Huang. “Searching in Internet of Things: Vision and Challenges.” In International Symposium on Parallel and Distributed Processing with Applications, 0:201– 6. Los Alamitos, CA, USA: IEEE Computer Society, 2011. doi:10.1109/ISPA.2011.53.

[6] Liu, Yuxi, and Guohui Zhou. “Key Technologies and Applications of Internet of Things.” In 2012 Fifth International Conference on Intelligent Computation Technology and Automation (ICICTA), 197–200, 2012. doi:10.1109/ICICTA.2012.56.

[7] Aggarwal, Charu C., Naveen Ashish, and Amit Sheth. The Internet of Things: A Survey from the DataCentric Perspective, Managing and Mining Sensor Data, 2013.

[8] Jung, Jaeyeon, Balachander Krishnamurthy, and Michael Rabinovich. “Flash Crowds and Denial of Service Attacks: Characterization and Implications for CDNs and Web Sites.” In Proceedings of the 11th International Conference on World Wide Web, 293– 304. WWW ’02. New York, NY, USA: ACM, 2002. doi:10.1145/511446.511485.

[9] Dunkels, A., B. Gronvall, and T. Voigt. “Contiki - a Lightweight and Flexible Operating System for Tiny Networked Sensors.” In 29th Annual IEEE International Conference on Local Computer Networks, 2004, 455–62, 2004. doi:10.1109/LCN.2004.38.

[10] Heddeghem, Ward Van. “Cross-Layer Link Estimation For Contiki-Based Wireless Sensor Networks.” Vrije Universiteit Brussel, 2009.

[11] Osterlind, F., A. Dunkels, J. Eriksson, N. Finne, and T. Voigt. “Cross-Level Sensor Network Simulation with COOJA.” In Proceedings 2006 31st IEEE Conference on Local Computer Networks, 641–48, 2006. doi:10.1109/LCN.2006.322172.

[12] Dunkels, Adam. “Rime - a Lightweight Layered Communication Stack for Sensor Networks.” Delft, The Netherlands, 2007.

[13] Roberts, E., Bassett, B. and Lochner, M., 2020. Bayesian anomaly detection and classification for noisy data. *International Journal of Hybrid Intelligent Systems*, pp.1-16.