FORENSIC ANALYSIS ON DISTRIBUTED DENIAL OF SERVICE ATTACK ON IOT ENVIRONMENT

[1]HAFIZUDDIN SHAHRIL FADZIL, [2,]ZUL-AZRI IBRAHIM, [3,]FIZA ABDUL RAHIM, [4]SAIFUL AMIN SHARUL NIZAM, [5]HARIS ISKANDAR MOHD ABDULLAH, [6]MUHAMMAD ZULHUSNI MUSTAFFA

[1,4,5,6]UNITEN R&D Sdn. Bhd., Selangor, Malaysia

[2]College of Computing and Informatics, Universiti Tenaga Nasional, Malaysia

[2,3]Institute of Informatics and Computing Energy, Universiti Tenaga Nasional, Malaysia

[3]Razak Faculty of Technology and Informatics, Universiti Teknologi Malaysia, Malaysia

E-mail: [1]hafizuddin.shahril@uniten.edu.my, [2]zulazri@uniten.edu.my, [3]fiza.abdulrahim@utm.my, [4]saiful.amin@uniten.edu.my, [5]haris.iskandar@uniten.edu.my, [6]zulhusni@uniten.edu.my

ABSTRACT

The Distributed Denial of Service (DDoS) attack is a malicious attempt to render the users unable to access a server service, usually by temporarily disabling or suspending its hosting server services. With the increase popularity of IoT devices such as the massive deployment of smart meter in Advance Metering Infrastructure, can create a situation where attacker can launch a DDoS attack in this environment. This work will focus on analyzing the impact of DDoS attack in AMI by performing data analysis from DDoS attacks that performed from IoT testbed. The testbed is use as a platform to perform the testing using multiple variation of DDoS attacks that can be launch from IoT devices. It also helps the system detect any DDoS attacks against IoT devices by tracking any abnormalities in the communication inside the testbed and connected IoT devices.

Keywords: *Internet of Things (IoT), IoT testbed, Distributed Denial of Service (DDoS).*

1. INTRODUCTION

The Internet of Things (IoT) is a recent evolution that is increasingly reshaping our future in communication technology. This technology allows small embedded devices to connect and interact, improving the capacity of such devices to serve human needs better [1]. IoT will be the gateway to the future technical solution for several industries, including energy generation, manufacturing, agriculture, and health care [2].

For example, in the field of power generation, IoT can be used to track and manage smart meters and provide working power station staff with reports and alarms quickly. The use of these systems guarantees the supply of electricity to consumer houses and essential facilities, such as the healthcare center. The different uses of IoT in relieving traffic, controlling smart lighting, monitoring noise, managing waste, and even shaping stronger and safer building structures were explored in several studies [3].

As a result of the potential benefits to the government, its people, and even the environment, the idea of smart cities is starting to emerge. IoT allows different physical objects or things such as sensors, actuators, mobile phones, and so on to connect and communicate with each other to accomplish a common purpose to implement the above vision [4].

However, many of them have weaknesses that make them vulnerable to different attacks due to the limited capabilities of IoT devices. Regardless of its security level, a vulnerable IoT system can be a dangerous hole in any network [5]. Many attacks, including Distributed Denial of Service (DDoS) attack, Main-In-The-Middle (MITM) attack, spoofing attack, impersonation attack, have included exploiting the vulnerabilities of IoT devices.

There has also been a spike in botnet attacks. One of popular past case is the Mirai botnet; this botnet will infected IoT devices by manipulating default credentials [6]. According to Proofpoint, baby monitors, smart TVs, smart light bulbs, and other smart home devices were more than 25% of the botnet's target [7]. Hundreds of IoT devices have been compromised and driven to launch Denial of Service (DoS) attacks on vital servers. Network Time Protocol (NTP) and Domain Name Service (DNS) are used by these attacks as a type of Distributed Denial of Service (DDoS) attack. One study stated that the main reason why the Mirai botnet is so successful is that hundreds of individuals use easy to install, low-cost IoT devices produced with little or no regard for security protection whatsoever [8].

Distributed Denial of Service (DDoS) attack is an attempt to disrupt the normal network traffic of a targeted website, server, service, and network resource by overwhelming the target with a flood of malformed or request packets sent from multiple devices [9]. This will cause the target to not be able to reply to request packets from legitimate users, which will affect the availability of the target. As DDoS attack is difficult to detect because of the characteristics of it traffic that used the same traffic as the normal communication make the attempt to block these attack a big challenge. This attack aim to deplete the resources such as Central Processing Unit (CPU), memory and bandwidth available in the server that offer services to clients.

The testbed that was developed in this project aim to provide an analysis on the impact DDoS attack in IoT environment which in this case the AMI and the result of the analysis will be discussed in the finding.

2. LITERATURE REVIEW

2.1 Distributed Denial of Service Attack

A Denial of Service (DoS) is an attack that is conducted to make the resources of networks and devices inaccessible to legitimate users so that no one else can access them [10]. Attackers will create a scenario in which the organizations will come to a grinding halt. These attacks are primarily targeted at personal computers, default computers, web servers, etc. [11].

Most attackers will attempt to compromise three aspects of information security: confidentiality, integrity, and availability of information. Confidentiality is compromised when the attacker discovers a way to gain access to the information. Integrity is compromised when the attacker gains access to information in order to modify and alter it. Availability is compromised when an attacker can block the information from being access by legitimate users.

Novice attackers cannot compromise the confidentiality and integrity of the information because that requires them to gain authorized access to the information, which is very difficult to do. Thus, they will attempt to target the availability of the information because they do not need any authorized access to do that. Many common DoS attacks depend on the vulnerabilities of TCP/IP protocols. UDP Flood, ICMP Flood, TCP SYN Flood and HTTP Request Flood are few of the typical DoS attacks. To initiate these attacks, attackers either make use of single computers or multiple computers.

DDoS attacks can be challenging to identify since zombies can be located across the globe. As a consequence, it is not possible to separate them from legitimate traffic. Common Denial of Service (DoS) attack involved an attacker attemp to take down the target computer by flooding it with Internet Control Message Protocol (ICMP) echo request, also known as ping request to initiate the ICMP flood or Ping flood attacks [12]. The victim's network is flooded with request packets, with the expectation that the network will respond with an equal number of response packets. This puts a strain on the network's incoming and outgoing channels, consuming a significant amount of bandwidth and causing a DoS. Ping requests are typically used to determine the connectivity of two computers by calculating the round-trip time from the time an ICMP echo request is sent to the time an ICMP echo response is received. They are, however, used to flood a target network with data packets during an attack.

Another form of Denial of Service (DoS) attack is the TCP SYN flood or SYN flood [13]. This attack takes advantage of the usual Transmission Control Protocol (TCP) three-way handshake to drain the resources on the targeted server and make it unresponsive. In general, a SYN flood DDoS attack happens when the attackers send TCP request packets faster than the targeted machine can process. This is done by using multiple machines to send multiple request packets simultaneously, causing the targeted network to slow down and finally become unresponsive to reply requests from legitimate users.

The client sends the SYN request packet first to the server using the three-way handshake in a normal connection. Then the server will respond to the client with the SYN-ACK packet to approve the client's request. The client will respond to the server with the ACK packet to confirm that the server has accepted its request. In the TCP SYN flood, the attacker repeatedly sends the SYN request packet first to the targeted port on the targeted server using the three-way handshake, sometimes using a fake IP address.

Then the server, unaware of the attacks, receives several requests to create a connection, apparently legitimate. Each attempt will be replied with a SYN-ACK packet from the targeted port to the attacker to acknowledge the attacker's request. This will require the server to use its resources to open the attacker’s connection and wait for the attacker to recognize its SYN-ACK packet for some time. The server will not close the connection by sending an RST packet during this time, and the connection stays open. Another SYN packet would arrive before the link can be disabled. This leaves an increasingly large number of half-open connections.

TCP SYN flood attacks are often known as “half-open” attacks. Finally, when the server’s connection overflow tables fill up, service to legitimate clients would be refused. While the above described “classic” SYN flood attempts to exhaust network ports, SYN packets can also be used in DDoS attacks that attempt to block the target pipe with fake packets to slow down the network. The TCP SYN flood is also known for its spoofed IP or fake IP that prevents the server and the forensic team from recognizing the computer of the real attacker.

The HTTP flood is a sort of Denial of Service (DoS) attack that uses seemingly valid GET or POST Hypertext Transfer Protocol (HTTP) requests to assault a web server or application [14]. HTTP flood assaults are volumetric attacks that often use a zombie army botnet, which is a collection of connected computers that has been maliciously taken over, usually with the help of software such as a Trojan horse.The purpose of the attack to slow down the attacked site’s bandwidth, and the magnitude of the attack is calculated in bits per second or how many the attacker can send in a second.

HTTP flood is a complex layer seven or application layer attack that uses no malformed packets, spoofing, or reflection methods and needs less bandwidth to bring down the targeted site or server than the other attacks. When an HTTP client, such as a web browser, establishes a connection with a server or application, it sends an HTTP request, which is normally one of two types: GET or POST. POST requests are used to access dynamic resources, while GET requests are used to get conventional, static information like images.

The attack is most successful when it leads the server or application to assign the maximum available resources to each and every request. As a result, the attacker will usually try to overwhelm the server or application by sending several requests, each of which is as processing-intensive as possible. From the attacker's standpoint, a POST request, which includes parameters that necessitate complicated server-side processing, looks to be the most resource-effective for this purpose.

Table 1 shows the summary of the studies performed to determine the type of DDoS attack. The table lists the DDoS type, which layer in the Open System Interconnection (OSI) model the DDoS operates in, protocol the DDoS exploited, and how they exploited it.

Table 1: DDoS Attack Summary

DDoS	Category		
	OSI Layer	*Protocol exploited*	*Exploitation*
ICMP Flood	Layer 3 (Network layer)	ICMP	Rapidly sends ICMP echo request to the target
TCP SYN Flood	Layer 4 (Transport Layer)	TCP	Establish a half-open connection with the server by exploiting the TCP handshake process
HTTP Flood	Layer 7 (Application layer)	HTTP	High GET or POST request rates from the attacker compared to the target

We used these three types of DDoS attacks in this study because these attacks are common DDoS attack types used by attackers such as the Mirai botnet to attack their target, such as a server. We also choose these attacks based on the different protocols these attacks may exploit and the different Open System Interconnection (OSI) layers that these attacks operate in. That way, we can monitor various attacks that use different protocols and operating on different OSI layers. We can also monitor what effect these attacks have on the IoT environment.

2.2 IOT Testbed

A significant step in product development is checking the security of IoT devices before they are released to the market, and this is an area in which testbeds can be extremely useful [15]. A security testbed is a predefined testing environment that monitors all devices, triggers, attacks, and tests [16].

First, researchers in [2] introduced the FIT IoT-LAB testbed, an open experimental testbed on a broad scale. The FIT IoT-LAB testbed is the type of testbed that simulates a large IoT network spectrum such as hardware, topologies, OS, protocols stack, and libraries used. The testbed uses OpenWSN, which implements a complete stack of protocols based on IoT standards, including IPv6, 6TiSCH, 6LoWPAN, UDP, RPL, and CoAP. The computers and robots in the testbed can be assigned to different topologies, including star topology and mesh topology.

Second, researchers in [17] presented the WHYNET (Wireless Hybrid NETwork) testbed. The WHYNET testbed is the type of testbed that covers wireless network protocol such as channel used, usage patterns, traffic, and mobility of wireless devices. The testbed conduct experiments to test the network performance between real device nodes and TWINE emulated device nodes using TCP, UDP, and HTTP network protocols. The physical components and nodes in the simulator can be set in various topologies such as mesh topology and star topology.

Third, researchers in [18] describe the deployment and experimentation architecture of an IoT experimentation testbed deployed at Santander city in Spain called the SmartSantander testbed. The testbed is a type of testbed covering data mapping as the testbed collects various data using various sensor devices. The testbed components communicate and send data on an IEEE802.15.4 network that is a technical standard for Low-Rate Wireless Personal Area Networks (LR-WPANs) using the HTTP protocol. Sensors that are in the testbed are organized into a mesh network.

In this study, we have developed a testbed to simulate a real IoT device environment. This testbed uses star topology as its network topology to connect all the IoT devices. The network standards used in this testbed are Ethernet connection and WLAN connection. The network protocols used to send data in this testbed are ICMP, TCP, and HTTP.

The summary of reviews on the existing testbed and our developed testbed is shown in *Table 2*. The table lists the testbed type, network standards used, network protocol used, and network topology. *Table 2* also compares the other testbed with the proposed testbed in this study.

Table 2: Testbed Summary

Testbed	Category			
	Testbed type	*Network Standard*	*Network Protocol*	*Network Topology*
FIT IoT-LAB	IoT Simulation	IEEE802.15.4(LR-WPAN)	IPv6, 6TiSCH, 6LoWPAN, UDP, RPL, CoAP	Star, Mesh
WHYNET	Network Protocol	IEEE802.11(WLAN) Cellular network	TCP, UDP, HTTP	Star, Mesh
SmartSantander	Data mapping	IEEE802.15.4(LR-WPAN)	HTTP	Mesh
Developed Testbed	IoT Simulation	IEEE 802.3 (Ethernet) IEEE 802.11 (WLAN)	ICMP, TCP, HTTP	Star

3. RELATED WORK

DDoS attack has been one of the most serious threats in digital era. The effect of DDoS attack can be devastating because the attack damages the components and make it impossible to provide normal services. It is quite complicated to simulate DDoS attack. Thus, researchers need to understand the network topology and experiment aspects in order to conduct a complicated DDoS attack experiment. The experiment required aspects in attack, background traffic, network topology, defense technology, testing and data collecting, outcome assessment.

There are many researchers have designed a testbed and simulate DDoS attack. Paper [19] has built a testbed to design a platform to test and evaluate DDoS attack defense program. The testbed simulated based on the reality of the most common network structure design. There are 3 background traffic terminals set in the experiment. The traffic set to sending FTP/TCP normal flow. Each link bandwidth of 1Mbit/s. 3 linked to pass the same routing node to its destination. The routing capability is 100 packets. Droptail, a mechanism for regulating the queue length of network nodes, is used in the experiment. The link bandwidth is 3 Mbit/s, with a 50ms routing link delay. The CBRl UDP attack traffic flow is sent by three terminals. They also configured the attack traffic and packet size to guarantee that the routing node cache queue was occupied within the 200ms pulse length. As a result, TCP traffic exhibit the initial time loss, followed by four queue overflows with increasing time intervals. Meanwhile, the gadget gathers and records three retransmission timeout RTOs of l.06s, 2AS, and 4.23s.

Flooding assaults and vulnerability attacks are the two main types of DDOS attacks. Vulnerability exploitation in the Ruby XML parser is described in Paper [20] . The attack sends a stream of invalid Web service request payloads to the Ruby server, each carrying a deeply nested meaningless XML message (up to 100,000 levels deep). The payload is approximately 1.5 MB in size. Each of the XML messages sent will be attempted to be loaded by a susceptible XML parser. Memory usage and CPU usage are monitored using SNMP. Usage before attack and during attack are recorded.

DETER is an advanced testbed facility available in the market that enable researchers to conduct experimentation on cybersecurity and educational exercise. Using DETER for DDoS experiments have explored dynamics and effects of DDoS attacks on complex networks [21]. Paper [22] did experiment on DDoS using DETER testbed. An attacker clients will conduct a DDoS packet flooding assault against the victim server. They used FLAT, PULSE, and RAMP distributions to create UDP and TCP floods for assaults in various scenarios. The Server throughput will be harmed by the attack traffic. The attack traffic is forwarded to the intended destination via intermediate node. As a result, the server will received connection that contains lawful traffic requested by legitimate node as well as attack traffic launched by the attacker node.

Several attack scenarios can be created and simulated using the testbed such as HTTP flooding, ICMP attack, TCP SYN, and UDP flood. Paper [23] developed three attack scenarios which is UDP Flood, HTTP-GET/POST, and TCP-SYN. In UDP Flood, the message size varies from 512 to 1024 bytes and is transmitted every 0.01 to 0.05 seconds. As a result, all of the malicious users will send 20 to 100 packets of data to the victim server per second. The victims' servers were unable to function as a result of the UDP attack. Two distinct programmes were installed in the browsers of the general and malicious users for HTTP-GET and HTTP-POST. On the bad clients, an attack application called was deployed, while on the general clients, the general HTTP browser application was installed. The assault programme appears to be a general server, but it actually targets the victim server. The malicious client sends the SYN packet to the server first in the TCP-SYN attack. After the bad client receives the SYN + ACK packet from the server, it sends the SYN packet instead of ACK packet. The dataset features are shown in Table 3.

Table 3: Dataset Features

No.	Value	Name	Description
1	10.0.0.98	SRC ADD	Source IP Address
2	10.0.0.26	DES ADD	Destination IP Address
3	2664	PKT ID	Identify of Packet
4	1033	FROM NODE	Identify of Lower Layer
5	1018	TO NODE	Identify of Hugh Layer
6	17	PKT TYPE	Type of Packet
7	614	PKT SIZE	Packet size
8	NULL	FLAGS	Flags of Packet
9	NULL	FIP	Identify of Transfer Layer
10	NULL	SEQ NUMBER	Sequence Number
11	4	NUMBER OF PKT	Number of Received Packet
12	3269	NUMBER OF BYTE	Number of Received Bytes
13	Encap	NODE NAME FROM	Name of Low Layer
14	ip	NODE NAME TO	Name of High Layer
15	1	PKT IN	Input Packet
16	0	PKT OUT	Output Packet
17	0	PKTR	Routing Packet
18	0	PKT DELAY NODE	Delay occurred at host
19	190.292	PKT RATE	Rate of packet receive
20	155,516	BYTE RATE	Rate of bytes receive
21	817.25	PKT AVG SIZE	Average received packet size
22	1	UTILIZATION	Used packet

Existing DoS research has primarily focused on determining denial of service using legitimate traffic metrics. The following parameters are widely used: (a) packet loss, (b) traffic throughput, (c) delay in request-response, (d) transaction duration, and (e) resource allocation [24]. Paper [25] did experiment of DDoS attack using two different testbeds namely QUT and MPLS testbeds. SSH for commands and SFTP or SCP for file transfer are the communication methods used by both testbeds. SYN floods, ICMP floods, and UDP floods are examples of simple flood-type assaults that can be fed 'onto the wire' at fast speeds. At the victim end, the source IP address and TTL value are extracted, and an IP-address-to-hop-count database is built.

This paper aims to design a testbed that consist of Raspberry Pi which used to act as smart meter. The Rapberry Pi is an IoT device that able to be programmed and easily configured. Then, another vulnerable component of AMI such as data collector and server are simulated using virtual machine which installed in a laptop. In paper [26], the writer create a testbed using a laptop running Ubuntu to simulate an attacker device,. The Smart Galaxy Watch (SGW) was used as a victim wearable device in the testbed (VWD). The SGW and VWD are connected to the same WiFi. The simulation was performed by using BoNeSi. The attack packets are UDP, TCP, AND ICIMP flooding packets and using Wireshark for analysis and record attack packet details. The BoNeSi programme has an additional option called 50k-bot, which is generated at random by exploiting 50,000 IP addresses.

4. METHODOLOGY

This paper reviews experiments that study on DDoS attack including in different environment. However, this paper focus on study of cyber-attack occur in the smart grid environment especially on components of advanced metering infrastructure (AMI). Following to existing study, components of AMI that vulnerable to cyber attacks include smart meter, data collector, and meter data management system (MDMS). This paper also reviews experimental result from other journals to gain knowledge on designing testbed to simulate the cyber-attack. This paper experiments the DDoS attack by designing a testbed to simulate the connection between components of smart grid. Then, this paper conducts an analysis based on the simulation of DDoS attack.

To study the cyber-attack, a testbed required to be designed to simulate the DDoS attack and analyze the attack. Those vulnerable components are simulated using raspberry pi, wireless access point, and a laptop. Raspberry pi's acted as smart meters. Virtual machine is installed in the laptop to act as data collector and meter data management system.

First, raspberry pi's are configured to act as smart meters. The codes are obtained from GitHub and were amended based on the objective of this paper. The pi's is configured to send data mimic like power consumption to the laptop that configured with a Virtual Machine that acted as Data Collector to received data from the pi's.

The operating system used in the laptop is Linux. The laptop contains high volume of RAM and storage. Wireshark is readily installed in the laptop to be used in analysis of cyber-attack. NetSim software is used to monitor the connection between components of testbed is configure. This paper used NetSim because it is easier and faster to configure compared to other open source simulator like for example NS3. Virtual machine was installed in the laptop used UBUNTU operating system to generate data sent to the data collector. All incoming data from the smart meter is stored in an MYSQL database.

4.1 Testbed Development

In this study, the IoT testbed consists of three main components: hardware components, software components, and network components. Figure 1 shows the testbed topology that consists of four Raspberry Pis with the IP address Smart Meter A (192. 168.43.111), Smart Meter B (192. 168.43.112), Smart Meter C (192. 168.43.113), and Smart Meter D (192. 168.43.114) respectively are connected to the access point, and the access point is connected to the laptop with the IP address (192.168.43.120). The four Raspberry Pis acted as smart meters and sends data to the laptop that acted as a data collector through the access point.

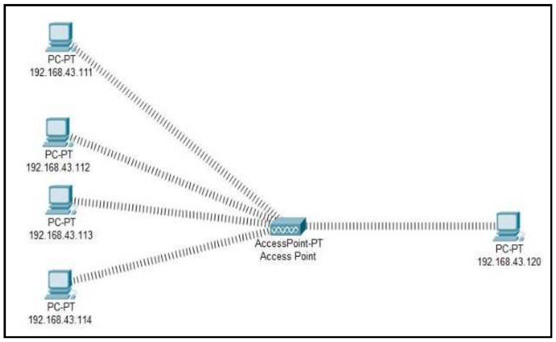


Figure 1: Testbed Topology

Smart Meter B, Smart Meter C, and Smart Meter D will launch the DDoS script to affect the testbed data collector. Smart Meter B, Smart Meter C, and Smart Meter D will launch multiple packets to the data collector simultaneously. This will create a flood of data that will flood the network traffic connecting the smart meters to the data collector. The data collector will be forced to receive the flood of data coming from Smart Meter B, Smart Meter C, and Smart Meter D, which causes the data collector to be unable to receive the measurement data from another legitimate smart meter, Smart Meter A. It is expected that the data collector will be unable to send measurement data to the server collector, thus interrupting the data transmission process.

Packet file analysis will be performed to detect network intrusions and other suspicious activity. Packet capture or PCAP is a valuable resource for file analysis and monitoring network traffic. Packet collection tools like Wireshark can be used to collect network traffic and translate it into a human-readable format.

4.2 Attack Development

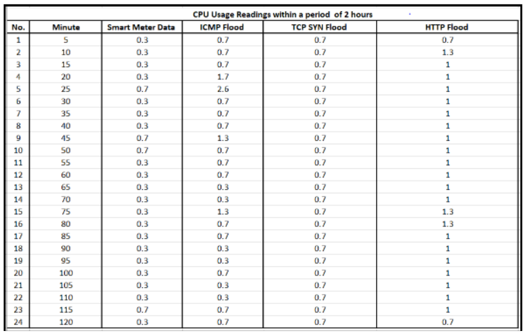
To launch the attack, this paper obtained DDoS attack code from GitHub. The codes were amended according to the objective. Four smart meters are used to simulate the DDoS attack to increase the result of attack in the Wireshark. The experiment starts with a normal traffic to simulate a smart meter as a normal client communicating to data collector. Then, the experiment continued with turning on DDoS attack script using three raspberry pi's which were configured as smart meter.

Two sessions of testing were executed where the first testing involved of running a normal topology data transmission where the smart meter will send data to the data collector. In the second testing, the DDoS was executed using three Raspberry Pis as the attackers with the aim to deny one normal Raspberry Pi sending traffic to the data collector.These experiments were conducted to analyse the network behavior during regular and attack traffic and the analysis will be focused on the impact of the attack to the following resources in the data collector server:

- CPU usage of the data collector (Figure 2)
- Memory usage of the data collector (Figure 4)
- Time taken by the data collector to reply ping request (Figure 6)
- Smart meter's throughput (Figure 9)

5. DISCUSSION OF FINDINGS

Figure 2 shows a spreadsheet table that records the CPU usage of the data collector during normal smart meter data transfer and during all of the DDoS attacks being executed in a wireless network in a period of 2 hours. The table shows that at minute 5, the CPU usage was 0.3 percent when smart meter data were being transferred. When ICMP flood, TCP SYN Flood, HTTP Flood attack was being executed, the CPU usage increase to 0.7 percent, meaning ICMP Flood, TCP SYN Flood, and HTTP Flood did affect the CPU usage in the data collector. Sometimes, ICMP Flood and HTTP Flood increase CPU usage to 2.6 percent and 1.3 percent. This means that these attacks contributed to an increase in CPU usage.



CPU Usage Readings within a period of 2 hours

No.	Minute	Smart Meter Data	ICMP Flood	TCP SYN Flood	HTTP Flood
1	5	0.3	0.7	0.7	0.7
2	10	0.3	0.7	0.7	1.3
3	15	0.3	0.7	0.7	1
4	20	0.3	1.7	0.7	1
5	25	0.7	2.6	0.7	1
6	30	0.3	0.7	0.7	1
7	35	0.3	0.7	0.7	1
8	40	0.3	0.7	0.7	1
9	45	0.7	1.3	0.7	1
10	50	0.7	0.7	0.7	1
11	55	0.3	0.7	0.7	1
12	60	0.3	0.7	0.7	1
13	65	0.3	0.3	0.7	1
14	70	0.3	0.3	0.7	1
15	75	0.3	1.3	0.7	1.3
16	80	0.3	0.7	0.7	1.3
17	85	0.3	0.7	0.7	1
18	90	0.3	0.3	0.7	1
19	95	0.3	0.3	0.7	1
20	100	0.3	0.7	0.7	1
21	105	0.3	0.3	0.7	1
22	110	0.3	0.3	0.7	1
23	115	0.7	0.7	0.7	1
24	120	0.3	0.7	0.7	0.7

Figure 2: CPU usage readings comparison within 2 hours

Figure 3 shows the graph for CPU usage, which the blue line representing smart meter data transfer is the lowest one on the graph. This means that smart meter data have the lowest CPU usage in the graph. The grey line represents TCP SYN Flood increase higher than the blue line meaning that TCP SYN flood has a higher CPU usage than the smart meter data. The orange line represents ICMP Flood that increases higher than the blue and grey line and sometimes higher than the yellow lines meaning ICMP Flood has a higher CPU usage than the TCP SYN flood and smart meter data. Sometimes, ICMP Flood increases the CPU even higher than the HTTP flood. Finally, the yellow line represents HTTP flood had increased the higher among the lines meaning that HTTP flood had the most effect on the CPU usage.

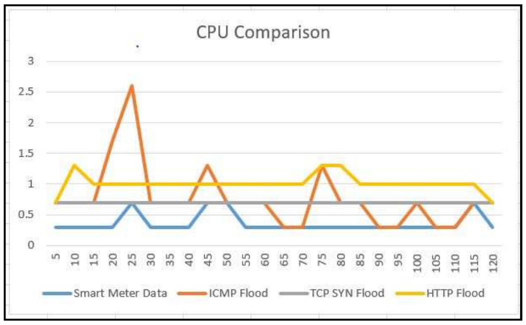
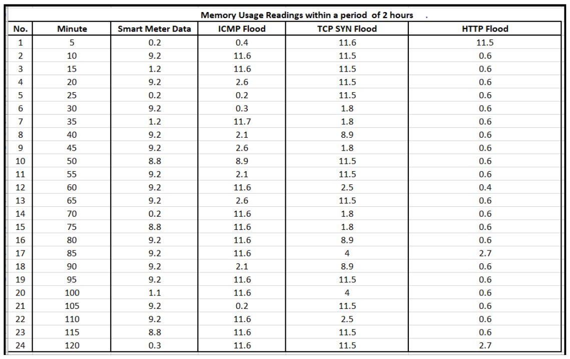


Figure 3: CPU usage readings comparison

Figure 4 shows a spreadsheet table that records the memory usage of the data collector during normal smart meter data transfer and during all of the DDoS attacks being executed in a wireless network in a period of 2 hours. The table shows that at minute

5, the memory usage was 0.2 percent when smart meter data wear being transfer. When the ICMP flood attack was being executed, the memory usage was 0.4 percent and then increased to 11.6 percent in minute 10 meaning ICMP had a higher effect on memory usage than the smart meter data.



Memory Usage Readings within a period of 2 hours

No.	Minute	Smart Meter Data	ICMP Flood	TCP SYN Flood	HTTP Flood
1	5	0.2	0.4	11.6	11.5
2	10	9.2	11.6	11.5	0.6
3	15	1.2	11.6	11.5	0.6
4	20	9.2	2.6	11.5	0.6
5	25	0.2	0.2	11.5	0.6
6	30	9.2	0.3	1.8	0.6
7	35	1.2	11.7	1.8	0.6
8	40	9.2	2.1	8.9	0.6
9	45	9.2	2.6	1.8	0.6
10	50	8.8	8.9	11.5	0.6
11	55	9.2	2.1	11.5	0.6
12	60	9.2	11.6	2.5	0.4
13	65	9.2	2.6	11.5	0.6
14	70	0.2	11.6	1.8	0.6
15	75	8.8	11.6	1.8	0.6
16	80	9.2	11.6	8.9	0.6
17	85	9.2	11.6	4	2.7
18	90	9.2	2.1	8.9	0.6
19	95	9.2	11.6	11.5	0.6
20	100	1.1	11.6	4	0.6
21	105	9.2	0.2	11.5	0.6
22	110	9.2	11.6	2.5	0.6
23	115	8.8	11.6	11.5	0.6
24	120	0.3	11.6	11.5	2.7

Figure 4: Memory readings comparison within 2 hours

When the HTTP Flood attack is being executed, the memory usage was 11.5 percent, but then it decreases to 0.6 percent and remains at that level for some time. This means that the attack affected the memory performance at first, but it no longer had an effect. This could be because the data collector needs to use a lot of memory to process the HTTP packet, but afterward, the data collector did need a lot of memory to process the HTTP packets. When the TCP SYN Flood attack was being executed, the memory usage increase to 11.6 percent like ICMP flood, meaning that the attack had the same effect on the memory just like ICMP flood.

Figure 5 shows the graph for memory usage, in which the blue line represents smart meter data transfer. The orange line representing ICMP flood and the grey line representing TCP SYN Flood is higher than the blue lines, meaning that TCP SYN Flood and ICMP flood had more effect on memory usage than the smart meter data. Finally, the yellow line represents HTTP flood had decreased the lowest among the lines meaning that HTTP flood had the least effect on the CPU usage.

Figure 6 shows a spreadsheet table that records the time taken for the data collector to reply to ping requests during normal smart meter data transfer and all of the DDoS attacks being executed in a wireless network in 2 hours. The table shows that at minute 5, the time taken to reply to ping was 3.89 milliseconds when smart meter data were being transferred—the time taken increased sometimes to 8.82 in minute 10. When the ICMP flood attack was being executed, the time taken was 4.41 milliseconds meaning the ICMP flood did not increase the time taken for the data collector to reply to the ping request. When the TCP SYN Flood attack is being executed, the time taken was 13.6 milliseconds.

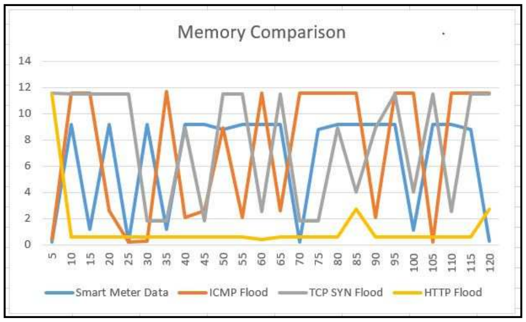
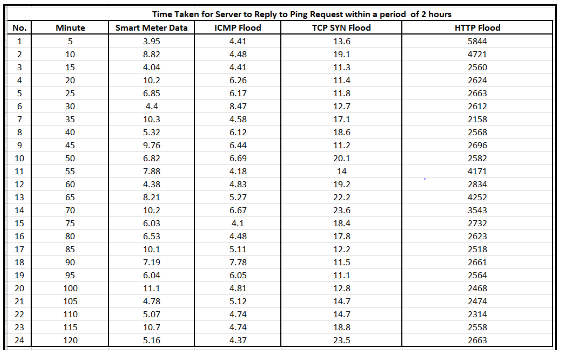


Figure 5: Memory usage readings comparison within 2 hours

The attack did increase the time taken for data collector to reply to ping requests compared to ICMP and smart meter data. When the HTTP Flood attack was being executed, the time taken to reply to ping request was 5844 milliseconds meaning that the attack did increase the time taken to reply to ping request even longer than ICMP, TCP SYN Flood, and smart meter data. This means that this attack had the most effect on delaying the data collector from replying to ping requests.



Time Taken for Server to Reply to Ping Request within a period of 2 hours

No.	Minute	Smart Meter Data	ICMP Flood	TCP SYN Flood	HTTP Flood
1	5	3.95	4.41	13.6	5844
2	10	8.82	4.48	19.1	4721
3	15	4.04	4.41	11.3	2560
4	20	10.2	6.26	11.4	2624
5	25	6.85	6.17	11.8	2663
6	30	4.4	8.47	12.7	2612
7	35	10.3	4.58	17.1	2158
8	40	5.32	6.12	18.6	2568
9	45	9.76	6.44	11.2	2696
10	50	6.82	6.69	20.1	2582
11	55	7.88	4.18	14	4171
12	60	4.38	4.83	19.2	2834
13	65	8.21	5.27	22.2	4252
14	70	10.2	6.67	23.6	3543
15	75	6.03	4.1	18.4	2732
16	80	6.53	4.48	17.8	2623
17	85	10.1	5.11	12.2	2518
18	90	7.19	7.78	11.5	2661
19	95	6.04	6.05	11.1	2564
20	100	11.1	4.81	12.8	2468
21	105	4.78	5.12	14.7	2474
22	110	5.07	4.74	14.7	2314
23	115	10.7	4.74	18.8	2558
24	120	5.16	4.37	23.5	2663

Figure 6: Comparison for the time taken for the server to reply to ping request within 2 hours

Figure 7 and Figure 8 show the graph for the time taken to reply to ping requests. The graph shows that the blue line representing smart meter data transfer and orange line representing ICMP flood are at the same level, meaning that the ICMP flood did not increase the time taken to reply to ping requests. The grey line representing TCP SYN Flood is higher than the blue and orange lines, meaning that TCP SYN Flood increased the time taken to reply to ping compared to ICMP flood and smart meter data. Finally, the yellow line represents HTTP flood had increased the highest among the lines meaning that the attack did increase the time taken to reply to ping requests. This means that this attack had the most effect on delaying the data collector from replying to ping requests.

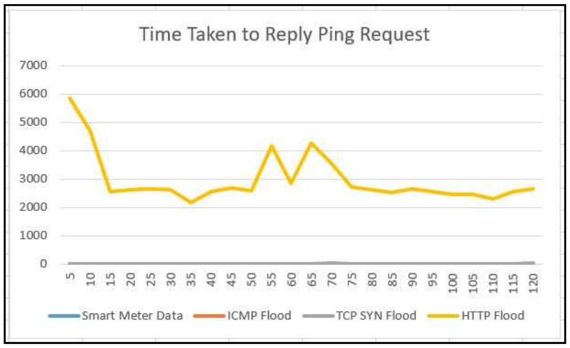


Figure 7: Comparison for the time taken for the server to reply to ping request within 2 hours

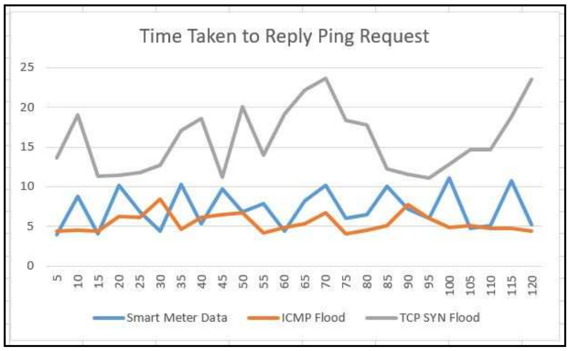
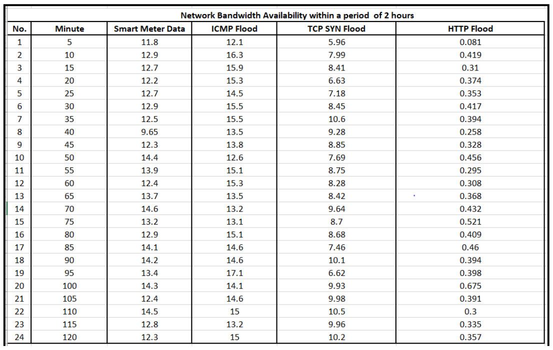


Figure 8: Comparison for the time taken for the server to reply ping request within 2 hours for smart meter data, ICMP flood and TCP SYN flood.

Figure 9 shows a Microsoft Excel table that records the network bandwidth of the data collector during normal smart meter data transfer and all of the DDoS attacks being executed in 2 hours. The table shows that at minute 5, the network bandwidth was 11.8 Mbit/sec when smart meter data were transferred. When the ICMP flood attack was being executed, the memory usage was 12.1 Mbit/sec, meaning the ICMP flood did not decrease the network bandwidth. When the TCP SYN Flood attack is being executed, the network bandwidth was 5.96 Mbit/sec meaning the attack did decrease the network bandwidth. When the HTTP Flood attack was being executed, the network bandwidth decreases to 0.081 Mbit/sec, meaning that the attack did reduce the network bandwidth the most. This means that this attack had the most effect to decrease network bandwidth.

Thus, it takes many very capable devices in terms of CPU performance and memory to process data very fast and launch a successful DDoS attack on the target. Additionally, as mentioned in the literature review, HTTP floods require less bandwidth than other attacks to bring down the targeted devices. Thus, it explained why HTTP flood DDoS attack shows an effective result in lowering the bandwidth availability of the network in the wired and wireless network.



Network Bandwidth Availability within a period of 2 hours

No.	Minute	Smart Meter Data	ICMP Flood	TCP SYN Flood	HTTP Flood
1	5	11.8	12.1	5.96	0.081
2	10	12.9	16.3	7.99	0.419
3	15	12.7	15.9	8.41	0.31
4	20	12.2	15.3	6.63	0.374
5	25	12.7	14.5	7.18	0.353
6	30	12.9	15.5	8.45	0.417
7	35	12.5	15.5	10.6	0.394
8	40	9.65	13.5	9.28	0.258
9	45	12.3	13.8	8.85	0.328
10	50	14.4	12.6	7.69	0.456
11	55	13.9	15.1	8.75	0.295
12	60	12.4	15.3	8.28	0.308
13	65	13.7	13.5	8.42	0.368
14	70	14.6	13.2	9.64	0.432
15	75	13.2	13.1	8.7	0.521
16	80	12.9	15.1	8.68	0.409
17	85	14.1	14.6	7.46	0.46
18	90	14.2	14.6	10.1	0.394
19	95	13.4	17.1	6.62	0.398
20	100	14.3	14.1	9.93	0.675
21	105	12.4	14.6	9.98	0.391
22	110	14.5	15	10.5	0.3
23	115	12.8	13.2	9.96	0.335
24	120	12.3	15	10.2	0.357

Figure 9: Network bandwidth availability readings comparison within 2 hours

6. CONCLUSION

From the knowledge obtained during the studies on the multiple types of IoT testbed developed by other researchers, we designed and implemented an IoT testbed to test multiple types of DDoS attacks. Every type of DDoS attack had its advantages and disadvantages that can be manipulated by the researchers and see the impact each DDoS attack had on the IoT environment wired and wireless.

ACKNOWLEDGEMENT

This study was funded by Tenaga Nasional Berhad Seed Fund (U-TD-RD-19-25) in collaboration with TNB Asset Management Department. We would like to thank UNITEN R&D Sdn. Bhd. for fund management.

REFRENCES

[1] V. A. Memos, K. E. Psannis, Y. Ishibashi, B. G. Kim, and B. B. Gupta, “An Efficient Algorithm for Media-based Surveillance System (EAMSuS) in IoT Smart City Framework,” *Futur. Gener. Comput. Syst.*, vol. 83, no. 2018, pp. 619–628, 2018, doi: 10.1016/j.future.2017.04.039.

[2] C. Adjih *et al.*, “FIT IoT-LAB: A large scale open experimental IoT testbed,” *IEEE World Forum Internet Things, WF-IoT 2015 - Proc.*, pp. 459–464, 2015, doi: 10.1109/WF-IoT.2015.7389098.

[3] A. Zanella, N. Bui, A. Castellani, L. Vangelista, and M. Zorzi, “Internet of things for smart cities,”

IEEE Internet Things J., vol. 1, no. 1, pp. 22–32, 2014, doi: 10.1109/JIOT.2014.2306328.

[4] L. Atzori, A. Iera, and G. Morabito, "The Internet of Things: A survey," *Comput. Networks*, vol. 54, no. 15, pp. 2787–2805, 2010, doi: 10.1016/j.comnet.2010.05.010.

[5] O. Badve, B. B. Gupta, and S. Gupta, "Reviewing the Security Features in Contemporary Security Policies and Models for Multiple Platforms," pp. 479–504, 2016, doi: 10.4018/978-1-5225-0105-3.ch020.

[6] C. Kolias, G. Kambourakis, A. Stavrou, J. Voas, and I. Fellow, "DDoS in the IoT," *Computer (Long. Beach. Calif).*, vol. 50, no. 7, pp. 80–84, 2017, doi: 10.1109/MC.2017.201.

[7] C. Stergiou, K. E. Psannis, B. G. Kim, and B. Gupta, "Secure integration of IoT and Cloud Computing," *Futur. Gener. Comput. Syst.*, vol. 78, pp. 964–975, 2018, doi: 10.1016/j.future.2016.11.031.

[8] J. A. Jerkins, "Motivating a market or regulatory solution to IoT insecurity with the Mirai botnet code," *2017 IEEE 7th Annu. Comput. Commun. Work. Conf. CCWC 2017*, 2017, doi: 10.1109/CCWC.2017.7868464.

[9] N. M. Zahri, Z. A. Ibrahim, F. A. Rahim, and Y. Yusoff, "Experimental study on software-defined network implementation for ddos attack detection and mitigation," *Int. J. Adv. Trends Comput. Sci. Eng.*, vol. 9, no. 4, pp. 6074–6081, 2020, doi: 10.30534/ijatcse/2020/278942020.

[10] N. Tripathi and B. Mehtre, "DoS and DDos Attacks: Impact, Analysis and Countermeasures," *Proc. Natl. Conf. Adv. Comput. Netw. Secur.*, no. July, pp. 1–6, 2013.

[11] R. V. Deshmukh and K. K. Devadkar, "Understanding DDoS attack & its effect in cloud environment," *Procedia Comput. Sci.*, vol. 49, no. 1, pp. 202–210, 2015, doi: 10.1016/j.procs.2015.04.245.

[12] M. Bogdanoski and A. Risteski, "Wireless network behavior under ICMP ping flood DoS attack and mitigation techniques," *Int. J. Commun. Networks Inf. Secur.*, vol. 3, no. 1, pp. 17–24, 2011.

[13] M. Bogdanoski, T. Shuminoski, and A. Risteski, "Analysis of the SYN Flood DoS Attack," *Int. J. Comput. Netw. Inf. Secur.*, vol. 5, no. 8, pp. 15–11, 2013, doi: 10.5815/ijcnis.2013.08.01.

[14] K. Singh, P. Singh, and K. Kumar, "Application layer HTTP-GET flood DDoS attacks: Research landscape and challenges," *Comput. Secur.*, vol. 65, pp. 344–372, 2017, doi: 10.1016/j.cose.2016.10.005.

[15] O. Abu Waraga, M. Bettayeb, Q. Nasir, and M. Abu Talib, "Design and implementation of automated IoT security testbed," *Comput. Secur.*, vol. 88, 2020, doi: 10.1016/j.cose.2019.101648.

[16] P. Cao, E. C. Badger, Z. T. Kalbarczyk, R. K. Iyer, A. Withers, and A. J. Slagell, "Towards an unified security testbed and security analytics framework," *ACM Int. Conf. Proceeding Ser.*, vol. 21-22-Apri, pp. 1–2, 2015, doi: 10.1145/2746194.2746218.

[17] M. Varshney, Z. Xu, S. Mohan, Y. Yang, D. Xu, and R. Bagrodia, "Whynet: A Hybrid Testbed for Large-Scale, Heterogineous and Adaptive Wireless Networks.," p. 35, 2007, doi: 10.1145/1287767.1287775.

[18] L. Sanchez *et al.*, "SmartSantander: IoT experimentation over a smart city testbed," *Comput. Networks*, vol. 61, pp. 217–238, 2014, doi: 10.1016/j.bjp.2013.12.020.

[19] S. Ning and Q. Han, "Design and implementation of DDoS attack and defense testbed," *2012 Int. Conf. Wavelet Act. Media Technol. Inf. Process. ICWAMTIP 2012*, pp. 220–223, 2012, doi: 10.1109/ICWAMTIP.2012.6413478.

[20] D. Schmidt *et al.*, "A Distributed Denial of Service Testbed," *Int. Fed. Inf. Process.*, no. September, pp. 20–23, 2010.

[21] T. Benzel *et al.*, "Experience with deter: A testbed for security research," *2nd Int. Conf. Testbeds Res. Infrastructures Dev. Networks Communities, TRIDENTCOM 2006*, vol. 2006, no. January, pp. 379–388, 2006, doi: 10.1109/TRIDNT.2006.1649172.

[22] D. Kaur, M. Sachdeva, and K. Kumar, "Study of DDoS attacks using DETER Testbed," *Researchmanuscripts.Com*, vol. 3, no. 2, p. 13, 2012, [Online]. Available: http://www.researchmanuscripts.com/may2012/2.pdf.

[23] S. Alzahrani and L. Hong, "Generation of DDoS Attack Dataset for Effective IDS Development and Evaluation," *J. Inf. Secur.*, vol. 09, no. 04, pp. 225–241, 2018, doi: 10.4236/jis.2018.94016.

[24] J. Mirkovic *et al.*, "Towards user-centric metrics for denial-of-service measurement," *Proc. 2007 Work. Exp. Comput. Sci.*, no. June, pp. 13–14, 2007, doi: 10.1145/1281700.1281708.

[25] D. Schmidt and S. M. Shalinie, "An Investigation into the Detection and Mitigation of Denial of Service (DoS) Attacks," *An Investig. into Detect. Mitig. Denial Serv. Attacks*, 2011, doi: 10.1007/978-81-322-0277-6.

[26] Shwetarani, N. M. F. Qureshi, and D. R. Shin, "Performance analysis of IoT-enabled DDoS botnets in wearable devices," *J. Theor. Appl. Inf. Technol.*, vol. 99, no. 16, pp. 4026–4043, 2021.