EAI Endorsed Transactions
on Security and Safety

Research Article EAI.EU

Spill the Beans: Extrospection of Internet of Things by Exploiting Denial of Service

Vinay Sachidananda*, Suhas Bhairav, and Yuval Elovici

iTrust, Singapore University of Technology and Design, Singapore

Abstract

Internet of Things (IoT) exposes various vulnerabilities at different levels. One such exploitable vulnerability is Denial of Service (DoS). In this work, we focus on a large-scale extensive study of various forms of DoS and how it can be exploited in different protocols of IoT. We propose an attack and defense framework called OWL which is tailored for IoT and that can perform various forms of DoS on IP, Bluetooth, and Zigbee devices. We consider various DoS vulnerabilities such as illegitimate packet injection, Bluetooth Low Energy (BLE) scanning attack, Zigbee frame counter-attack, etc., regarding IP, Bluetooth and Zigbee devices. To understand how resilient is IoT for DoS, we propose two new metrics to measure the Resilience and the Quality of Service (QoS) degradation in IoT. We have conducted large-scale experimentation with real IoT devices in our security IoT testbed. The experiments conducted are for DoS, Distributed Denial of Service (DDoS) by setting up Mirai and Permanent Denial of Service (PDoS) using BrickerBot on various IoT devices. We have also compared our framework with the existing state of the art tools.

Received on 10 February 2019, accepted on 02 April 2019, published on 29 April 2019

Keywords: Security Analysis, Security and Privacy, Vulnerabilities, Denial of Service, Internet of Things

doi:10.4108/eai.13-7-2018.159803

1. Introduction

The Internet of Things (IoT) is increasingly becoming an integral part of everyone's lives. IoT devices like smart lights at home [1], motion sensors to detect movements [2], voice assistants to perform activities like playing music, providing weather updates [3] [4], and many more, are continuing to occupy almost every household. Though IoT devices are experiencing an exponential pace of adoption, they have various security loopholes making them vulnerable to numerous attacks.

As a case in point, IoT devices have been recently used to launch various attacks such as Denial of Service (DoS) and steal end-user information [5] [6] [7]. A website of a cybersecurity expert was brought down by a Distributed Denial of Service (DDoS) attack through IoT devices [8] and there was also a massive Internet outage along the East Coast in the United States using IoT devices [9]. Recently, Mirai malware had compromised a huge number of Deutsche Telekom routers [6] by performing a DDoS attack. There have been instances where it even resulted in Permanent Denial of Service (PDoS) by bricking the IoT devices [10].

Apart from exposing various vulnerabilities, IoT devices have less computing power compared to desktop computers and other computing devices and thus, are susceptible and less resilient to such attacks. The IoT devices handle a limited amount of traffic for performing basic applications. For example, a teardown of Amazon Echo reveals its hardware constituents [11]. It runs a Texas Instruments DM3725 Digital Media processor. It has a 250MB mobile DRAM made by Samsung and a 4GB storage chip. So, a comparison of a traditional IT device like a desktop that has a 500GB of storage, 16GB of RAM and an i3 processor with an Amazon Echo portrays that Amazon Echo's processing and storage capabilities are too low. In other words, DoS, DDoS, and PDoS attacks are a threat to IoT devices. The current state of the art lacks in detailed large-scale experimentation and study of various forms of DoS and the resilience of IoT against DoS that encompass the IP, Bluetooth, and Zigbee devices. In this work, we logically perform such study.

First, in our work, we propose our attack and defense framework called OWL (*Optimized Weighted Legitimates and Illegitimates*). OWL is tailored for IoT which can successfully perform DoS against IP, Bluetooth, and Zigbee IoT devices. OWL scans the

*Corresponding author. Email: sachidananda@sutd.edu.sg

IoT environment, analyzes, monitors and mutates the packets that will be accepted by the IoT device. OWL produces legitimate and illegitimate packets to perform the DoS. Not to forget, OWL also includes techniques in performing classical resource exhaustion for DoS. However, OWL stands out in performing DoS attacks through a few mutated packets by exploiting various DoS vulnerabilities of IoT devices. We have compared and evaluated OWL with two of the state-of-the-art DoS tools, LOIC (Low Orbit Ion Cannon) [12] and hping3 [13], where LOIC and hping3 can perform DoS only on IP based IoT devices and only through resource exhaustion. Next, we introduce a DoS and DDoS defense framework for IoT. The framework is capable of analyzing the network traffic to determine if there is a DoS or a DDoS attack on a specific IoT device. Suppose there is an attack, the defense framework takes appropriate steps to mitigate the attack by changing the IP address of the IoT device and providing alerts for Bluetooth and Zigbee devices. On the other hand, if an IoT device within the network is launching a DoS attack on other devices, the defense framework will automatically disconnect the network connection of the attacking IoT device.

Second, we have introduced *IoT Resilience* (R_{IoT}) metric to evaluate the resilience of an IoT device against DoS, DDoS and PDoS. IoT Resilience will be calculated based on the services running on an IoT device and the security vulnerabilities exposed by the IoT device. Furthermore, we introduce *Quality of Service Degradation* (D_{IoT}) metric to measure the violation of various quality of service requirements for IoT. Furthermore, we also adopt legacy metrics such as throughput, allocation of resources and Normal Packet Survival Ratio.

Finally, we have carried out detailed experiments and evaluation of DoS, DDoS, and PDoS against IoT. We have performed DoS attacks through TCP connections [14] [15], SYN flooding, ICMP flooding [16] and other methods. We have performed Bluetooth Low Energy (BLE) scanning and Packet Injection attacks on Bluetooth devices. For Zigbee devices, we perform Identify Action Attack and Frame Counter Attack. We carry out DoS attacks through legitimate and illegitimate packets and evaluate the resilience of the IoT devices. We consider legitimate packets (normal) and illegitimate packets (mutated). Furthermore, we have used Mirai (Mirai malware forms a Botnet of IoT devices and tries to compromise various other devices connected to the network) to perform DDoS attacks within a controlled environment. Furthermore, we perform and evaluate a PDoS attack using BrickerBot on IoT devices. The experiments are carried out in a controlled environment in our IoT security testbed [17]. We have successfully conducted experiments on more than 69 IoT devices.

In this paper, our contributions are threefold:

- We introduce a new attack and defense framework called OWL, specially designed for IoT which can perform various forms of DoS attacks on IP, Bluetooth, and Zigbee devices. OWL will scan, analyze, monitor and mutate packets to perform attacks. OWL encompasses a completely automated defense framework.
- We introduce two new metrics to evaluate IoT Resilience and IoT QoS degradation.
- To understand the impact of DoS, DDoS and PDoS on IoT, we have performed extensive real-world experiments using our framework on more than 69 IoT devices in our IoT security testbed. Our experiments are extensive and detailed to provide a complete understanding of various forms of DoS attacks on IoT and its resilience and QoS on IoT.

1.1. Threat Model

The user is trustworthy and honest and can install various IoT devices with the network setup as described by the manufacturer. The credentials for remote access of our setup are not disclosed. All the IoT devices used in our setup follow the standard protocol specifications. The attacker is present outside the network setup. The attacker wants to accomplish a few goals of violating security requirements such as availability, confidentiality, etc.

The expertise of the attacker lies in eavesdropping the setup without having physical access to any of the IoT devices. The attacker can inject packets in wireless communication and can take control of the IoT devices and can cause service denial. Furthermore, the IoT devices can form a botnet based on the bots like Mirai, and the IoT devices can be remotely accessed by bots like BrickerBot to cause service denial.

The structure of the paper is as follows: Section 2 introduces our attack and defense framework. In Section 3, we introduce two new metrics and also discuss the adapted legacy metrics. In Section 4, we discuss our experimental methodology and setup. Section 5 provides experimental results and in Section 6, we discuss the related work. Finally, we conclude in Section 7 with future work.

2. OWL:Proposed Attack and Defense Mechanism

In this section, we propose our attack and defense framework OWL (***O**ptimized **W**eighted **L**egitimates and Illegitimates*). OWL framework is tailored for IoT and consists of three algorithms namely OWL orchestration, OWL Attack, and OWL Defense. OWL takes into consideration the IP, Bluetooth and Zigbee devices for performing various DoS attacks using legitimate and illegitimate packets. OWL can (a) scan, analyze and monitor the network traffic to understand the target IoT, (b) mutate packets that will be accepted by the IoT device, (c) exploit existing DoS vulnerabilities while injecting less traffic and (d) provides a defense mechanism for DoS attacks. OWL framework is evaluated on real IoT security testbed which we will explain in detail in Section 4.

Algorithm 1 OWL Orchestration

```
 1: procedure SCAN(addr)
 2:     device    scanAddrSpace(addr)
 3:     devType    fetchRepoData(device)
 4:     analyze(device)
 5: end procedure
 6: procedure ANALYZE(device)
 7:     pkt    createPkt(device)
 8:     storeRepo(pkt, device)
 9: end procedure
10: procedure MONITORNW(addr)
11:     monitorAndStorePacketsInDB(addr)
12:     calMaxima(addr)
13: end procedure
14: procedure CALMAXIMA(addr)
15:     pkts    getPktsFromDB(addr)
16:     maximaResults    fetchMaxima(pkts)
17:     storeMaximaResultsInDB(maximaResults)
18: end procedure
```

2.1. OWL Orchestration

OWL Orchestration *(Algorithm 1)* facilitates the scanning, analysis, monitoring and maxima calculation for all the IoT devices. The *Scan* procedure *(Algorithm 1: Line 1-5)* scans the address space for all the IoT devices in the environment and identify the IoT device type. The *Analyze* procedure *(Algorithm 1: Line 6-9)* creates a new packet based on the IoT device and stores it in the repository. The *MonitorNW* procedure *(Algorithm 1: Line 10-13)* monitors the entire network and captures the communication between all the IoT devices and then calls the *CalMaxima* procedure *(Algorithm 1: Line 14-18)* which calculates the maximum packet usage value (incoming and outgoing) for each IoT device and stores it in the repository.

2.2. OWL Attack

OWL Attack *(Algorithm 2: Line 1-15)* is responsible for performing DoS attacks on IoT devices. The attack varies on the IoT device. If the IoT device is IP based, then resource exhaustion attack and mutation attack is performed. The resource exhaustion attack is carried out via legitimate packets from the repository while the mutation attack involves illegitimate packets created as a result of mutation. Concerning Bluetooth devices, scan attack and packet injection attack is carried out. For Zigbee devices, touch attack and frame counter-attack are carried out.

Algorithm 2 OWL Attack

```
 1: procedure ATTACK(packet, device)
 2:     packetType    getPacketType(packet)
 3:     if packetType is ip OR ble OR zigbee then
 4:         if packetType is ip then
 5:             resAttack(packet, device)
 6:             illegitimateMutate(packet, device)
 7:         else if packetType is ble then
 8:             scanAttack(packet, device)
 9:             pktInjAttack(packet, device)
10:         else if packetType is zigbee then
11:             touchAttack(packet, device)
12:             frmCntAttack(packet, device)
13:         end if
14:     end if
15: end procedure
```

Algorithm 3 OWL Defense

```
 1: procedure REALTIMEVALIDATION(addrArr)
 2:     addressList    getDeviceAddr(addrArr)
 3:     while TRUE do
 4:         for addr in addressList do
 5:             maxima    getMaximaResults(addr)
 6:             time ← 0
 7:             incomingPackets ← 0
 8:             outgoingPackets ← 0
 9:             while time < 100 do
10:                 inPkts    getPktCntToDev(addr)
11:                 outPkts ← getPktCntFromDev(addr)
12:                 time ← time+1
13:             end while
14:             if incomingPackets > maxima then
15:                 initiateDoSAlert(addr)
16:             end if
17:             if outgoingPackets > maxima && addr is IP then
18:                 removeFromNw(addr)
19:             end if
20:         end for
21:     end while
22: end procedure
23: procedure INITIATEDOSALERT(addr)
24:     if addr is ip then
25:         changeRoutingTable(addr)
26:     else
27:         popUpBleZigAlert(addr)
28:     end if
29: end procedure
```

2.3. OWL Defense

OWL Defense *(Algorithm 3: Line 1-29)* caters to the DoS and DDoS defense mechanism functionality. The defense framework performs real-time monitoring to identify anomalies in the traffic *(Algorithm 3: Line 1-22)*. Each device is monitored for time T seconds in the network. Where time T can vary and in our case we have set $T = 100$. If the incoming packets for an IoT device exceed a threshold (i.e., maxima calculated in Algorithm 1), then DoS alert is initiated. If it is an IP based device, then the IP address of the device is changed in the routing table. For a BLE or a Zigbee device, an alert is shown that the corresponding device is under a DoS attack *(Algorithm 3: Line 23-29)*. Similarly, if the outgoing packets for an IP device exceed its threshold, then the device is removed from the IoT network.

3. Resilience and QoS degradation

We propose two new metrics namely (a) IoT Resilience and (b) QoS-IoT Degradation metric to understand how resilient is an IoT against DoS, DDoS, and PDoS. We define these two new metrics because of the existing legacy metrics lacks to capture the IoT device's resiliency and service degradation.

3.1. IoT Resilience

Before we define the *Resilience* of an IoT device, we need to understand its *Permeance* [18]. We define *Permeance* of an IoT device against a DoS, DDoS or a PDoS attack as:

Definition 1(a): *The total number of packets an IoT device can service over a period when it is bombarded with attack packets before the IoT device fails to provide service.*

$$P_{IoT} = S * \frac{(P_n * P_a)}{T_{RRT}}$$

P_n represents the total number of normal packets. P_a represents the total number of attack packets. T_{RRT} represents the *Request Response Time* of the IoT device. S represents the *Resilience constant* specific to an IoT device vulnerability. In [17], the authors have done penetration testing for IoT devices and have identified a metric system for port scanning to rate the vulnerable ports of the IoT device. We make use of the same metric system to measure our constant S. The *Resilience constant* S varies as a function of the risk level of the scanned ports. The total number of open ports running specific services on each one of them indicates a possibility of those services being affected when the device is under a DoS, DDoS or PDoS attack. Higher the number of open ports, higher are the chances of the device being attacked. Keeping this in mind, the authors in [17] calculate the *Exploitability Score* for an IoT device. We use the same methodology to calculate the score of the IoT devices used in our experiments.

The unit of Permeance is p^2/s. From the definition of *Permeance*, we can define *Resilience* of an IoT device against a DoS attack as:

Definition 1(b): *The resilience of an IoT device is defined as the reciprocal of its permeance.*

$$R_{IoT} = \frac{1}{P_{IoT}}$$

R_{IoT} is the resilience of an IoT device whose unit is s/p^2.

3.2. Quality of Service Degradation of IoT

To derive the QoS-IoT degradation, it is important to understand that a user can experience varying service degradation (partial to full-fledge) in the case of DoS and DDoS attacks. This indeed leads us to first specify the applications that are available for IoT such as video, audio, motion, lights, etc. Furthermore, we can understand that the quality of the services can be varied and also completely deprived of the user. Hence, it is necessary to take into consideration the specific requirements of the QoS for IoT. Video streaming on an IP camera, music streaming on Amazon Echo, detecting the presence by motion sensors, etc., can be leveraged as the services of the IoT. We can now define a *service* provided by an IoT device as:

Definition 2: *A service is a specific high-level task provided by an application from an IoT device which is meaningful and requested by an end-user.*

Considering QoS requirements (*Packet Loss, Burst Level, Packet Jitter, Packet Delay and Bandwidth*) from the current state-of-the-art [19], we consider a service to be successful if it meets all our adapted QoS requirements from an underlying IoT device. If one of the requirements is not met, then the service is classified as degraded. We define QoS-IoT degradation as:

Definition 3: *A set of applications providing services in an IoT device and violating a set of quality requirement Q_{sr} leads to QoS-IoT degradation.*

Congruent with quality standards recommended by standardization organizations, i.e., ITU-T, IETF, etc., we assign the normalized QoS weighted values as: Packet

Loss = 0.5, Burst Level = 0.4, Packet Jitter = 0.3, Packet Delay = 0.2 and Bandwidth = 0.1.

During a DoS or a DDoS attack on an IoT device, when there is packet loss, the device may not be able to provide the desired results to the user. Hence, we place a higher emphasis on packet loss since it affects the eventual outcome in terms of the final response. We assign it a weighted value of 0.5 which is the highest amongst all the considered QoS requirements. When a traffic burst takes place, the response is still valid. So we assign a value of 0.4 that is slightly less than packet loss. We assume the packets are received in order but in bursts. In our case, we assume Packet Jitter to have a weighted value of 0.3. We assume Packet Jitter consists of several Packet Delays throughout the interval of a device being under an attack. Packet Delay has a value of 0.2 when there is one occurrence of a packet arriving late. Bandwidth has the least weighted value because even if the bandwidth is low, the device is still able to provide service or respond to requests.

We assume, in an ideal scenario, a device to have a Q_{sr} as 1. Now, a corresponding weighted value of QoS requirements is added to the default Q_{sr} value as and when it is violated. For e.g., when there is a packet loss during the transmission, Q_{sr} value becomes Q_{sr}+0.5. Furthermore, from the current state of the art [20] [21] [22] [23], we assign a weighted value for the threshold T as *0.5*.

Now, we can formulate the metric for QoS-IoT degradation as the total number of denied services and the measure of the severity of service denial.

$$D_{IoT} = \frac{T_{ds}\,(i) * (Q_{sr} - T)}{T}$$

T_{ds} represents the total number of denied services for an IoT device i. Q_{sr} represents the summation of violation of various QoS requirements and T represents the threshold.

3.3. Legacy Metrics

We identify and discuss an array of DoS metrics known as *Legacy metrics* [24] and utilize them to quantify the impact of such attacks on IoT devices. The legacy metrics provide a deeper insight towards specifying the resilience of IoT devices to these attacks. We have chosen some of the widely used metrics from the state-of-the-art and are as follows:

Throughput For an IoT device, the throughput continues to increase for requests from users. The throughput is defined as, *the total number of bytes transferred per unit time from source to the destination.*

$$Throughput = \frac{\Sigma_{i=0}^{n}\, PD}{\Sigma_{i=0}^{n}\, (PAT) - (PST)}$$

where, PD represents packet delivered, PAT represents packet arrival time and PST represents packet start time.

Allocation of resources Allocation of resources is defined as *the ratio of the bandwidth of legitimate traffic to the bandwidth of attack traffic.*

$$Allocation\ of\ resources = \frac{BLT}{BAT}$$

where, BLT represents bandwidth of legitimate traffic and BAT represents bandwidth of attack traffic.

Normal Packet Survival Ratio Normal Packet Survival Ratio (NPSR) is defined as *the ratio of legitimate packets delivered to the user to the total number of packets delivered.*

$$NPSR = \frac{PL}{PL + PA}$$

Where PL represents the number of legitimate packets and PA represents the number of attack packets.

4. Experimental Methodology

Our experiments are conducted in a real-world network topological setup to evaluate our proposed framework and also to calculate the Resilience and QoS degradation of IoT devices against various DoS attacks.

Experimental Setup: We have performed our experiments in IoT security testbed as shown in Figure 1. The testbed consists of various IoT devices which are IP, Bluetooth and Zigbee based devices. We have chosen 69 various kinds of IoT devices (including 26 devices used for the setup of Mirai experiment and 5 devices for BrickerBot experiment) to make sure that the framework would be adaptable to various resource-constrained IoT devices. We have three dedicated machines to run our experiments and all the three can speak to each other. Our OWL framework runs on these machines as follows: (1) Orchestrating Machine: this machine runs *Algorithm 1* of our framework. The framework is initiated by orchestrating machine which scans for the available IoT devices in the testbed and fetches the packet information from the repository to identify the device type. Next, the framework analyzes the device and creates the respective IoT device packet. Then, the entire network is monitored

Figure 1. Smart Home Setup and IoT Security Testbed

and the maximum number of packets arising in and out of the IoT device is calculated. (2) The Control and Communication Machine runs *Algorithm 2*, which is dedicated to various attacks. (3) The Analysis Machine runs *Algorithm 3* of our framework, which is dedicated to the defense purpose.

4.1. Denial of Service

For conducting DoS attacks we have used our OWL framework and also used state of the art tools (LOIC [12] and hping3 [13]) for the fair comparison. The reason for choosing LOIC and hping3 is due to their ability to perform various types of DoS attacks namely TCP, UDP, and HTTP in a sophisticated manner with a high magnitude of impact [25]. However, both the tools are chosen have their drawbacks when compared to our OWL framework for attack purpose regarding performance, attack methods and surface, holistic inclusion of IP, Bluetooth and Zigbee IoT devices. For *IP devices* we perform two attacks which are as follows:
(1) **Resource Exhaustion**: OWL framework performs resource exhaustion by sending spoofed legitimate packets to the IoT devices. E.g., The communication between the Android App on the mobile phone and an IP Camera are monitored and the Analysis Machine injects legitimate packets into the communication network channel of the IP camera. This kind of resource exhaustion is done for various IP based IoT devices in our testbed in a Man in the Middle attack fashion.
(2) **Mutation Attack:** OWL generates Illegitimate Packets to perform mutation attack. OWL can mutate various TCP, UDP, HTTP packets from the repository that can be accepted by IoT devices. The packets are generated in a fashion that it can cause a DoS attack by exploiting vulnerabilities of the IoT device, other than resource exhaustion. E.g., The DoS attack is carried out on TP-Link Cam through the following process: We monitored the traffic of TP-Link Cam while it was performing its routine activities. The type of attack was based on the type of open port and the packet type of the corresponding port. The mutated packet could exploit vulnerable port and the session hijacking vulnerability to bypass authentication and request for admin login continuously causing DoS.

On **Bluetooth Low Energy (BLE) devices**, we were able to carry out DoS attacks on devices such as Fitbit, Blood Pressure Monitor, etc. The attacks performed on BLE were:
(1) **BLE Scanning Attack** (CVE-2017-13211) which involves *OWL* sending out a large number of scanning requests to BLE devices. The entire process is carried out for 90 seconds. Within this time frame, the BLE devices failed to respond resulting in DoS.
(2) **BLE Packet Injection Attack** involves *OWL* bombarding the BLE devices with a large number of illegitimate BLE packets resulting in the devices being overwhelmed. Within specific time δ, all the BLE devices failed to respond.

On **Zigbee devices**, we were able to carry out DoS attacks on various Zigbee devices such as Philips Hue, Samsung SmartThings Hub, etc. Using OWL Attack, we performed two attacks:
(1) **Identify Action Attack** is exploited based on the vulnerability discovered by [26]. A Zigbee bulb blinks when an *identify request* is sent. *OWL* sends an *identify request* to a specific Zigbee device and sets the maximum duration value of 18 hours, 12 minutes, 14 seconds. As a result, it continues to blink. The user is left with the only option of physically shutting down the lights.
(2) **Frame Counter Attack** is performed by setting a large value for the frame counter for the Zigbee devices [26]. Thus, a genuine packet exchange will result in the packet being rejected causing DoS.

4.2. Distributed Denial of Service

Compromised IoT devices are capable of carrying out distributed denial of service(DDoS) attacks on other IoT devices, computers or services. One such way of facilitating a DDoS is via malware, such as Mirai [27] that is used in our experimentation.

Mirai turns networked devices into remotely controlled Bots and was first detected in August 2016 by the Whitehat malware research group MalwareMustDie [27]. The initial version of Mirai targets IoT devices running on open Telnet/SSH ports and those devices that have default usernames and passwords. Once the devices are infected, Mirai begins targeting other IoT devices by sending a large number of packets. This results in overwhelming the resources of the victim IoT devices.

In Figure 2, we notice that the Mirai Botnet operation requires a *Command and Control (C&C) server*, *Report*

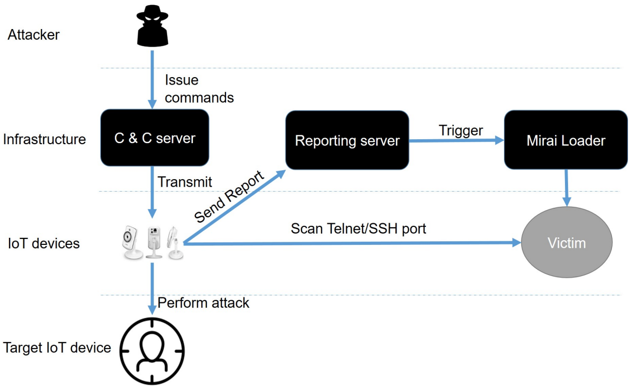


Figure 2. Mirai Setup

Server and *Loader*. When an attacker launches Mirai, it first scans for a potential victim. The Bot identifies devices containing Telnet or SSH connection. Then, the malware brute forces the credentials, infects the device and simultaneously sends the IP address of the device to the report server. The loader transmits the malware to infect the victim device. The infected device, in turn, begins similarly scanning for additional potential victims. At the same time, the C&C server issues command to attack various devices. There are different types of attack packets used by Mirai namely ACK, DNS, GREETH, GREIP, etc [28].

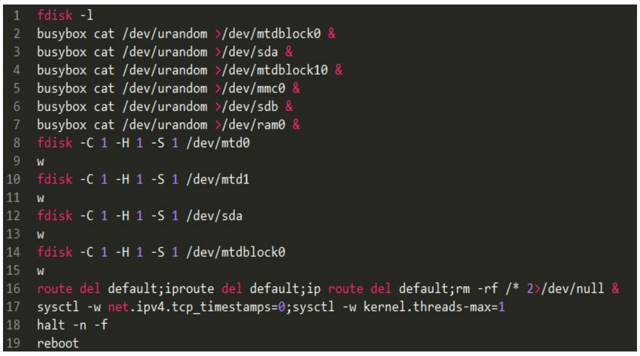
When we launched a DDoS attack using twenty-five (25) IP cameras on the victim IP camera, we observed that the VSE attack was the one with the highest throughput. This was followed by UDP and GREETH. The worst performing attack was DNS and ACK attack which generated a throughput of 4.2 Mbps and 5.9 Mbps respectively. In our Botnet experimentation, we used a total of twenty-six (26) D-Link DCS-942L [29] IP cameras, two laptops, and a dedicated access point. We monitored the network traffic on a desktop computer through a mirror port.

We created a Botnet comprising of twenty-five (25) D-Link DCS-942L IP cameras. The IP cameras were reset to their default *username* and *password*. The *Telnet* port of the twenty five (25) cameras was opened by issuing an *HTTP* command. A Mirai malware instance was then installed on twenty-five (25) cameras. Next, two laptops were used as servers. One laptop served as the *C&C* server while the other as the *Reporting server*. The IP address of the two servers was changed in the Mirai malware before being installed on the cameras. Then, we directed twenty-five (25) IP cameras, through the *C&C* server to attack the victim IP camera. As soon as the camera stopped streaming the video on the smartphone through the Android application, the DDoS attack was stopped by the defense framework.

4.3. Permanent Denial of Service

Permanent Denial of Service (PDoS) involves sabotaging an IoT hardware by exploiting its security flaws. The security flaws allow accessing IoT devices remotely and provide the ability to execute commands that perform various actions including system-level operations. The attack involves the execution of potentially harmful commands that modify or corrupt an IoT device's firmware, thus rendering it useless, as the IoT device loses its ability to boot or function. BrickerBot is a malware having the ability to carry out a PDoS attack [30].

BrickerBot is a malware that attacks IoT devices that run a specific version of the DropBear SSH server and target Linux devices running *Busybox* (usually IP cameras). The malware removes the default gateway, limits the kernel threads to one and disables timestamps of TCP. It deletes the boot loader and file system consisting of the Linux kernel. Once the file system has been deleted, the IoT device is unable to reboot [31]. Figure 2 shows the sequence of commands of the BrickerBot malware [32].



```
fdisk -l
busybox cat /dev/urandom >/dev/mtdblock0 &
busybox cat /dev/urandom >/dev/sda &
busybox cat /dev/urandom >/dev/mtdblock10 &
busybox cat /dev/urandom >/dev/mmc0 &
busybox cat /dev/urandom >/dev/sdb &
busybox cat /dev/urandom >/dev/ram0 &
fdisk -C 1 -H 1 -S 1 /dev/mtd0
w
fdisk -C 1 -H 1 -S 1 /dev/mtd1
w
fdisk -C 1 -H 1 -S 1 /dev/sda
w
fdisk -C 1 -H 1 -S 1 /dev/mtdblock0
w
route del default;iproute del default;ip route del default;rm -rf /* 2>/dev/null &
sysctl -w net.ipv4.tcp_timestamps=0;sysctl -w kernel.threads-max=1
halt -n -f
reboot
```

Figure 3. BrickerBot Command Sequence

When BrickerBot malware ran on an IoT device, the entire file system was wiped out. This resulted in the denial of all the QoS services, so Q_{sr} value was set to 0.5 +0.4 +0.3 +0.2 +0.1 =1.5. We assumed the IP camera consists of two services namely, video and audio streaming services on the smartphone android application and web application. As a result, T_{ds} was assigned a value of 2 (i.e. the denied services are audio and video). The QoS-IoT degradation was calculated as 4. The IoT resilience would be immaterial because the value is zero for a PDoS attack. In our experimentation, we used five (5) D-Link DCS-942L IP cameras to test BrickerBot. We reset the IP cameras and opened their Telnet port through a *HTTP* command. We placed the BrickerBot malware on the two cameras and remotely executed the malware through a shell script from a laptop. The BrickerBot malware had completely wiped out the file system on the cameras and the cameras failed to start. The LED on the cameras which was green when the camera was switched on had turned red and the Telnet communication between the laptop and the cameras failed. There was no traffic arising from the cameras after the attack. We tried to restart the cameras but observed that the LED behind the cameras remained red and failed to boot.

5. Experimental Results

In this section, we evaluate our OWL framework and other tools with our proposed metrics and provide a detailed analysis of all the results induced. Table 1 provides the comprehensive results of legacy metrics and resilience of all the IoT devices. Table 2 provides the Quality of Service and Degradation Results of all the IoT devices.

5.1. Denial of Service

From Table 1, we can infer that when OWL was used to perform DoS attacks, the throughput of the attack was low for all the devices. We noticed that OWL was able to completely bring down all the devices at a much faster rate compared to LOIC and hping3. E.g., to cause DoS on a Belkin Smart Switch, a throughput of 479.1 Mbps was required by OWL compared to 1055 Mbps as generated by LOIC.

We can infer that the allocation of resources for OWL took far less attack traffic bandwidth and more legitimate traffic to cause DoS in IoT devices. E.g., for Smart Things Hub to go down due to DoS, OWL had an allocation of resources value of 0.012 while LOIC had 2.6E-04 and hping3 had 2.9E-04.

OWL had a higher NPSR rate and required far less number of legitimate and attack packets compared to the other two tools. For example, NPSR value for Samsung Smart TV under OWL was 0.64 while the other two tools had lower values.

The resilience of IoT devices was less when OWL was used, compared to the other two tools as shown in Table 1. For example, Amazon Echo had a resilience of 9.7E-09 s/p^2 when OWL was used while the same Amazon Echo had a resilience value of 7.35E-08 s/p^2 and 9.35E-08 s/p^2 for LOIC and hping3 respectively.

From Table 2, we can infer that voice assistant device such as Amazon Echo, Echo Dot, and Google Home incurred a significant service degradation as there was a huge amount of packet loss. When a DoS attack was performed while the devices were playing music, the music streaming stopped. When the DoS attack stopped, after a few seconds, there was a sudden burst of music being played. The time interval of δ in which the music streamed (play and stop) varied, resulting in jitter. Also, there was a significant amount of delay during music streaming. As a result, Q_{sr} of the voice assistant devices was set to 1.4. The QoS-IoT degradation (D_{IoT}) value was 1.8. When IP cameras such as Nest cam, OmniGuard Camera, HK Vision Camera, Netatmo Camera, DLink, Withings Camera, Logi Circle, ARD Camera, and HK Vision Camera-2 were subjected to a DoS attack, a portion of video during the attack was not stored on the cloud and the video streaming stopped. We also noticed that, on some occasions, there was a burst of video for some periods during the attack. They also experienced loss of packets, jitter, and delay during streaming. As a result, Q_{sr} of the IP cameras were set to 0.5+0.4+0.3+0.2=1.4 and D_{IoT} value was 3.6. Similarly, the HP Printer was subjected to a DoS attack, the scanned copy failed to reach the email address. As a result, Q_{sr} of HP Printer was set to 0.5+0.4+0.3+0.2=1.4 and D_{IoT} value was 1.8. We could successfully perform DoS on various other IoT devices as mentioned.

Concerning Bluetooth devices, only OWL framework was able to successfully carry out DoS attacks. E.g., Fitbit-1 failed to send updates to the Android app when the throughput of the DoS attack reached 96.28 Mbps during a packet injection attack. We assumed two services being denied namely real-time transmission and collection. All the QoS requirements were violated. As a result, Q_{sr} was set to 0.5 + 0.4 + 0.3 + 0.2 + 0.1 = 1.4. Hence, D_{IoT} value was 3.6. Concerning Zigbee devices, GE Link Zigbee bulb failed to respond after the frame counter value was exceeded due to a large number of packets being sent. Thus, the throughput of 268.42 Mbps. The bulb failed to respond to commands sent via the Android app. As a result, all the QoS services were denied. The D_{IoT} value was 3.6. Similar values were observed across all the Bluetooth and Zigbee devices.

Analysis: The reason for the allocation of resources to be high in all the cases for OWL is because it picks the right legitimate packet from the repository and injects them after mutation. OWL can send a lower number of illegitimate packets in addition to more number of legitimate packets compared to the other two DoS tools. Hence, the throughput for OWL is lower than the other tools. NPSR value for OWL is higher compared to the other two tools because of its ability to send a higher number of intelligently modified legitimate packets.

5.2. Distributed Denial of Service

When we launched a DDoS attack using twenty-five (25) IP cameras on the victim IP camera, we observed that the VSE attack was the one with the highest throughput. This was followed by UDP and GREETH.

The worst performing attack were DNS and ACK attack which generated a throughput of 4.2 Mbps and 5.9 Mbps respectively. We found that ACK and DNS attacks have the highest allocation of resources value followed by SYN attack. ACK attack has an allocation of resources value of 0.668 while DNS attack has the value of 0.663. The SYN attack has a value of 0.305. The other attacks have low allocation of resources value.

Device	Throughput (Mbps)			Allocation			NPSR			Resilience		
	LOIC	*OWL*	*hping3*	*LOIC*	*OWL*	*hping3*	*LOIC*	*OWL*	*hping3*	*LOIC*	*OWL*	*hping3*
Echo	2.72E+05	5.83E+04	9.1E+05	5.16E-05	3.96E-04	1.4E-04	1.17E-04	9.11E-04	1.2E-04	7.35E-08	9.70E-09	9.35E-08
Nestcam	6.9E+04	5.59E+05	7.1E+04	0.58	1.99	6.6E-05	0.38	0.48	3.1E-04	2.10E-12	1.86E-12	4.17E-12
HP Printer	1.07E+05	7.34E+04	2.4E+05	1.32E-04	9.94E-04	3.1E-04	1.91E-05	0.02	1.3E-03	3.06E-09	2.98E-10	4.96E-09
Samsung TV	3.1E+05	2.09E+04	4.5E+05	1.93E-02	0.13	4.89E-03	0.22	0.64	0.09	8.01E-06	8.96E-08	9.89E-06
Wink Hub	1.18E+05	2.6E+04	2.3E+05	1.91E-06	0.03	1.98E-02	1.5E-06	5.14E-02	1.8E-06	2.82E-08	1.04E-11	2.95E-08
OmniGuard Cam	5234.1	632.7	6790.2	0.00021	0.0018	0.00024	0.0027	0.04	0.0029	5.24E-07	2.56E-07	6.37E-07
Google Home	5.9E+05	5.8E+04	6.5E+05	1.2E-04	1.9E-04	1.8E-04	1.6E-03	7.9E-02	2.9E-03	2.09E-08	1.43E-08	2.89E-08
DLink Camera	951.2	772.9	1004.8	1.7E-04	3.9E-04	2.1E-04	2.8E-03	0.9	2.9E-03	5.15E-09	1.24E-09	7.89E-09
Smart Things	1.04E+05	1.8E+04	1.7E+06	2.6E-04	0.012	2.9E-04	1.7E-04	0.81	2.8E-04	9.07E-10	4.99E-11	10.78E-10
HK Vision	1822.3	698.7	2987.7	2.1E-04	2.9E-04	2.3E-04	3.09E-03	0.42	3.78E-03	4.20E-13	1.49E-13	4.71E-13
Netatmo	2065.8	970.3	2897.8	3.6E-04	1.14E-03	3.9E-04	6.6E-04	0.039	6.9E-04	4.21E-10	4.19E-11	5.98E-10
Withings	2255.4	643.2	2987.1	2.5E-04	1.6E-03	2.9E-03	4.2E-06	0.01	4.8E-06	3.48E-12	1.13E-13	3.89E-12
Logi Circle	7.9E+04	5.8E+04	8.8E+04	0.11	0.21	0.12	9.8E-03	0.49	10.89E-03	6.71E-07	2.90E-08	6.93E-07
Philips Hue	712.8	588.7	989.2	2.3E-04	4.7E-04	2.7E-04	7.9E-05	5.12E-02	8.1E-05	2.81E-09	1.76E-09	2.97E-09
iSmart Alarm	3523.3	970.3	5624.8	3.8E-04	1.14E-03	3.9E-04	6.6E-05	4.54E-02	6.9E-05	5.30E-09	4.35E-09	5.89E-09
Sense Mother	2.02E+05	661.7	2.9E+05	7.62E-05	1.75E-04	7.78E-05	2.02E-04	0.33	2.9E-04	2.31E-09	5.72E-10	3.62E-09
ARD-Camera	2396.9	884.7	3426.8	1.1E-04	9.6E-04	1.4E-04	9.9E-04	0.04	10.55E-04	2.77E-10	3.53E-11	3.11E-10
OnHub Router	3.37E+05	7.2E+04	4.89E+05	3.3E-04	9.01E-03	3.8E-04	4.95E-05	0.04	5.03E-05	9.43E-09	5.65E-09	9.61E-09
Google WiFi	4.9E+05	6.5E+04	5.5E+05	3.8E-04	1.21E-03	4.9E-04	0.09	0.31	0.23	6.70E-10	6.10E-10	6.90E-10
Echo Dot	2.5E+05	5.4E+04	2.9E+05	5.59E-05	4.01E-04	5.98E-05	8.4E-03	0.39	8.9E-03	7.88E-09	4.40E-10	8.12E-09
SmartWatch	4869.2	552.1	7865.1	2.18E-06	1.9E-04	2.34E-06	5.8E-03	0.86	6.92E-03	1.69E-08	7.00E-09	2.33E-08
Smart Switch	1055.0	479.1	1987.6	1.6E-04	1.5E-03	2.7E-04	1.04E-03	0.52	1.89E-03	7.62E-08	1.98E-08	7.89E-08
Motion Sensor	1419.7	515.6	1762.8	6.17E-06	8.26E-04	6.89E-06	7.9E-04	0.5	8.6E-04	1.27E-09	4.15E-10	2.78E-09
CrockPot	5856.1	698.3	6724.6	5.4E-05	2.1E-03	5.7E-05	8.1E-04	0.99	8.73E-04	6.21E-08	1.98E-10	7.23E-08
Philips TV	44525.1	20887.6	52678.3	1.6E-04	0.01	1.89E-04	5.9E-05	0.27	6.5E-05	5.65E-09	3.96E-10	5.9E-09
Smart Scale	6362.9	650.8	6728.2	2.69E-05	2.77E-04	2.78E-05	0.094	0.78	0.19	1.02E-07	7.68E-09	1.34E-07
HK Vision-2	2354.1	1013.1	2896.2	4.6E-04	1.9E-03	4.9E-04	9.9E-04	0.22	10.9E-04	5.29E-10	4.95E-10	5.82E-10
Smart WiFi Plug	2139.9	369.5	2256.8	9.38E-05	4.99E-04	9.45E-05	5.91E-06	0.27	5.98E-06	8.50E-10	5.77E-10	8.78E-10
Fitbit-1	-	96.28	-	-	4.4E-04	-	-	4.1E-03	-	-	2.5E-04	-
Fitbit-2	-	92.48	-	-	5.9E-04	-	-	5.5E-03	-	-	2.07E-04	-
BP Monitor	-	119.74	-	-	0.001	-	-	8.4E-03	-	-	8.18E-05	-
BLE Watch	-	72.54	-	-	1.07E-03	-	-	1.14E-02	-	-	1.64E-04	-
Doorbell	-	135.64	-	-	1.4E-03	-	-	1.1E-02	-	-	4.8E-05	-
Nespresso Prodigio	-	218.5	-	-	2.05E-03	-	-	1.09E-02	-	-	1.8E-05	-
Philips Hue Bulb	-	305.56	-	-	4.79E-03	-	-	2E-02	-	-	5.09E-06	-
Osram Lightify	-	365.34	-	-	3.33E-03	-	-	1.2E-02	-	-	5.7E-06	-
GE Link	-	268.42	-	-	9.04E-03	-	-	4.01E-02	-	-	3.3E-06	-
IKEA Tradfri	-	325.08	-	-	4.5E-03	-	-	1.8E-02	-	-	4.9E-06	-

Table 1. Comprehensive Experimental Results for Various Metrics

Device	Metrics			
	T_{ds}	Q_{sr}	T	D_{IoT}
Echo	1	1.4	0.5	1.8
Nestcam	2	1.4	0.5	3.6
HP Printer	1	1.4	0.5	1.8
Samsung TV	1	1.4	0.5	1.8
Wink Hub	3	1.4	0.5	5.4
OmniGuard Cam	2	1.4	0.5	3.6
Google Home	1	1.4	0.5	1.8
DLink Camera	2	1.4	0.5	3.6
Smart Things	3	1.4	0.5	5.4
HK Vision	2	1.4	0.5	3.6
Netatmo	2	1.4	0.5	3.6
Withings	2	1.4	0.5	3.6
Logi Circle	2	1.4	0.5	3.6
Philips Hue	3	1.4	0.5	5.4
iSmart Alarm	2	1.4	0.5	3.6
Sense Mother	2	1.4	0.5	3.6
ARD-Camera	2	1.4	0.5	3.6
OnHub Router	1	1.5	0.5	2
Google WiFi	1	1.5	0.5	2
Echo Dot	1	1.4	0.5	1.8
SmartWatch	1	1.4	0.5	1.8
Smart Switch	1	1.4	0.5	1.8
Motion Sensor	1	1.4	0.5	1.8
CrockPot	1	1.4	0.5	1.8
Philips TV	1	1.4	0.5	1.8
Smart Scale	1	1.4	0.5	0.4
HK Vision-2	2	1.4	0.5	3.6
Smart WiFi Plug	1	1.4	0.5	1.8
Fitbit-1	2	1.4	0.5	3.6
Fitbit-2	2	1.4	0.5	3.6
BP Monitor	2	1.4	0.5	3.6
BLE Watch	2	1.4	0.5	3.6
Doorbell	2	1.4	0.5	3.6
Nespresso Prodigio	2	1.4	0.5	3.6
Philips Hue Bulb	2	1.4	0.5	3.6
Osram Lightify	2	1.4	0.5	3.6
GE Link	2	1.4	0.5	3.6
IKEA Tradfri	2	1.4	0.5	3.6

Table 2. Quality of Service and Degradation Results

The NPSR was high during an SYN attack while it is the lowest for GREIP and GREETH attacks. This implies that during an SYN attack, the device under attack receives and responds to packets at a higher rate when compared to the other attack types. GREIP attack has the least NPSR value of 0.00023. The IP camera had the least resilience to VSE and UDP attacks. The IoT resilience value for VSE is $2.6E-10\ s/p^2$ and UDP is $3.8E-10\ s/p^2$. The IP camera shows a higher resilience compared to the other attacks for DNS attack type with a value of $1.6E-05\ s/p^2$.

Analysis: VSE attack has the highest throughput when compared to all others. Mirai malware encapsulates GRE packet header with a UDP packet before launching an attack. The victim camera is bombarded with a significant number of UDP packets. GREIP, VSE, UDP and UDPPlain attacks are carried out using UDP packets resulting in such high throughput. The device does not send out an acknowledgment when UDP packets are sent to it. The transfer rate of ACK, DNS and SYN packets is less compared to UDP. Thus, this is reflected in their throughput values of the attack types. All the other attacks are all UDP related attacks. As a result, the throughput is high for GREETH, GREIP, UDP, UDPPlain and VSE attack types. For ACK, DNS, and SYN, because the IP camera responds to these packets by sending a response packet, it takes more resources to bring down the device. As a result, more packets are required to bring down an IoT device for ACK, DNS and SYN compared to the other attack types. Since more number of VSE, GREETH, UDP, UDPPLAIN and GREIP packets are sent to the IoT device compared to the other types of attack, the device is least resilient to such overwhelming type of attacks and it goes down.

5.3. Permanent Denial of Service

When BrickerBot malware ran on an IoT device, the entire file system was wiped out. This resulted in the denial of all the QoS services, so Q_{sr} value was set to 0.5 + 0.4 + 0.3 + 0.2 + 0.1 = 1.5. We assumed the IP camera consists of two services namely, video and audio streaming services on the smartphone android application and web application. As a result, T_{ds} was assigned a value of 2 (i.e. the denied services are audio and video). The QoS-IoT degradation was calculated as 4. The IoT resilience would be immaterial because the value is zero for a PDoS attack.

5.4. Defense Framework Analysis

When we conducted successful DoS and DDoS attacks in our experiments, we were able to detect those attacks from our defense framework. The threshold value for each of the IP based devices had been calculated using OWL. When the threshold values were exceeded, the necessary steps were taken by the defense framework. First, the IoT device's IP was changed as per the defense framework functionality. Second, we observed that during a DDoS attack (in the case of Mirai), the attacker devices were immediately removed from the network. Also, the victim camera's IP was changed.

Our defense framework achieved True Positives accuracy of at least 95% in all the cases. Concerning False Positives, we found that the false positives were obtained in cases where the actual interaction between IoT devices increased due to the legitimate network traffic. An important observation that we were able to make from these results is that on almost all of the occasions, our framework detected the DoS and DDoS attacks correctly. Similar True Positive and False Positive results were seen for DoS attacks. As we perform traditional defense mechanisms the overhead of the defense framework is minimal and with limited resources.

6. Related work

A DoS is a well-known concept, we have classified the current state-of-the-art according to Mobile Ad-hoc Networks, Wired Networks, Peer-to-Peer Networks, Internet of Things and DoS tools.

Mobile Ad-hoc Networks (MANETs) Jhaveri et al. [33] survey DoS attacks on MANETs and propose methodologies to detect and prevent such attacks. The attacks also include Gray hole, Blackhole, and Wormhole attacks. Kannhavong et al. [34] provide various details of flooding attacks along with wormhole attacks, replay attacks and link spoofing attacks on MANETs. They also discuss the implementation of various countermeasures. Jawandhiya et al. [35] categorize attacks against MANETs into *Passive*, *Active*, and *Miscellaneous*. Passive attacks include Eavesdropping attacks and Traffic monitoring. The authors provide a comprehensive overview of Active attacks such as Jamming attack, Byzantine attack and Transport Layer attacks (SYN flooding). DoS attacks are classified as Miscellaneous where resource exhaustion is carried out in MANETs. Besides, sleep deprivation attacks and routing table overflow attacks are analyzed.

Wired Networks Zargar et al. [36] classify DDoS flooding attacks as well as their countermeasures. However, networks are limited to wired systems. DDoS attacks arising from Botnets such as IRC-based, P2P-based and Web-based are also discussed. Mirkovic et al. [18] [37] [38] [39] [40] [41] provides a comprehensive analysis of DDoS on a network including the metrics to measure the impact of DDoS attacks. Bhandari et al. [24] elaborate on the various metrics that could be used to evaluate the performance of DDoS attacks.

Peer-to-Peer Networks Peer-to-Peer (P2P) systems organized in an unstructured manner are capable of denying service to legitimate users when used maliciously. The exploitation of Gnutella-based systems in such a manner has been demonstrated by Athanasopoulos et al. [42]. A compromised or malicious node intelligently forces its peers to download content from the victim. Naoumov et al. [43] present two distinct scenarios of exploiting P2P systems to perform DoS attacks by creating a DDoS engine. One scenario involves a distributed index amongst various peers being poisoned while the other involves the routing table being poisoned. Qi et al. [44] perform query and data flooding of P2P systems to analyze the impact of DDoS attacks.

Internet of Things Perakovic et al. [45] analyze protocols like UDP, SYN, NTP, ACK and their impact on connected IoT devices. However, authors in [45] do not involve analysis of varying types of devices present and their resilience against DoS attacks. Mirai's functionalities and operations on IoT devices are discussed by Kolias et al. [46]. The communication sessions between the compromised IoT devices and the Bot servers are analyzed but the effects of several attack types are not discussed.

Denial of Service Tools In [47], a comparison of various DDoS tools including LOIC are presented, but they do not evaluate them in an IoT environment. Helalat et al. [48] discuss the ability of an HTTP attack in a progressive manner using OWASP Switchblade and Slowhttptest software. The analysis of the network under attack provides a deep insight into the behavior of systems using HTTP protocols. Farina et al. [49] propose a SlowBot Net architecture consisting of LOIC to evaluate traffic generation and discuss the impact on mobile devices.

Nevertheless, none of the aforementioned research evaluated various DoS attacks, Mirai Botnet and BrickerBot Malware through DoS metrics capable of quantifying the impact on IoT devices. Furthermore, the above-mentioned work lacks to measure the resilience of IoT against DoS attacks.

7. Conclusion and Future Work

In this paper, we demonstrated and evaluated various forms of DoS attacks on IoT devices. We have done an extensive and holistic study of IP, Bluetooth and Zigbee IoT devices against various DoS attacks. We implemented, demonstrated and compared our attack and defense framework called OWL. We proposed two new metrics to calculate the Resilience of IoT devices and to evaluate the degradation of QoS in IoT devices. We carried out DDoS using IP cameras within a sophisticated environment and discussed the results. Besides, we also carried out PDoS attacks on real IP cameras. We conducted intensive experimentation on our IoT security testbed using more than 69 real IoT devices. We have discussed and analyzed our results and compared OWL with the existing state of the art tools, LOIC and hping3. We intend to apply and calibrate these metrics in the future to develop various defense mechanisms for IoT against DoS attacks.

References

[1] Ur B, Jung J, Schechter S. *The current state of access control for smart devices in homes.* In Workshop on Home Usable Privacy and Security (HUPS) 2013 Jul 24. HUPS 2014.

[2] Tozlu S, Senel M, Mao W, Keshavarzian *A Wi-Fi enabled sensors for internet of things: A practical approach.* IEEE Communications Magazine. 2012 Jun;50(6):134-43.

[3] Earley S. *Analytics, machine learning, and the internet of things.* IT Professional. 2015 Jan;17(1):10-3.

[4] Kim KJ. *Interacting socially with the Internet of Things (IoT): effects of source attribution and specialization in humanâĂŞIoT interaction.* Journal of Computer-Mediated Communication. 2016 Oct 20;21(6):420-35.

[5] Mirai Botnet attacks IoT. [n. d.]. *Mirai Malware for IoT.* https://www.symantec.com. [Accessed 21-05-19].

[6] Bank Info Security. [n. d.]. *Distributed Denial of Service using Mirai.* https://www.bankinfosecurity.com. [Accessed 21-05-19].

[7] Mirai Botnet on Routers. [n. d.]. *Mirai Malware launches DDoS.* https://www.theregister.co.uk. [Accessed 22-05-19].

[8] Brian Krebs. [n. d.]. *Mirai-Attack-On-Krebs.* https://krebsonsecurity.com. [Accessed 22-05-19].

[9] East Coast attacks. [n. d.]. *DDoS attack hits east coast.* https://www.wired.com. [Accessed 22-05-19].

[10] Bricker Bot. [n. d.]. *Permanent Denial of Service using BrickerBot.* https://security.radware.com. [Accessed 22-05-19].

[11] CNET. [n. d.]. *Amazon Echo teardown gets inside the smart speaker powered by the cloud.* https://www.cnet.com. [Accessed 15-01-19].

[12] Infosec Institute. [n. d.]. *Low Orbit Ion Cannon.* http://resources.infosec institute.com. [Accessed 22-05-19].

[13] Linux. [n. d.]. *hping3 network tool in Linux.* https://linux.die.net. [Accessed 22-05-19].

[14] Kuzmanovic A, Knightly EW. *Low-rate TCP-targeted denial of service attacks: the shrew vs. the mice and elephants.* In Proceedings of the 2003 conference on Applications, technologies, architectures, and protocols for computer communications 2003 Aug 25 (pp. 75-86). ACM.

[15] Schuba CL, Krsul IV, Kuhn MG, Spafford EH, Sundaram A, Zamboni D. *Analysis of a denial of service attack on TCP.* In Proceedings. 1997 IEEE Symposium on Security and Privacy (Cat. No. 97CB36097) 1997 May 4 (pp. 208-223). IEEE.

[16] Peng T, Leckie C, Ramamohanarao K. *Survey of network-based defense mechanisms countering the DoS and DDoS problems.* ACM Computing Surveys (CSUR). 2007 Apr 12;39(1):3.

[17] Sachidananda V, Siboni S, Shabtai A, Toh J, Bhairav S, Elovici Y. *Let the cat out of the bag: A holistic approach towards security analysis of the internet of things.* In Proceedings of the 3rd ACM International Workshop on IoT Privacy, Trust, and Security 2017 Apr 2 (pp. 3-10). ACM.

[18] Mirkovic J, Reiher P. *A taxonomy of DDoS attack and DDoS defense mechanisms.* ACM SIGCOMM Computer Communication Review. 2004 Apr 1;34(2):39-53.

[19] Kim HJ, Choi SG. *A study on a QoS/QoE correlation model for QoE evaluation on IPTV service.* In 2010 The 12th International Conference on Advanced Communication Technology (ICACT) 2010 Feb 7 (Vol. 2, pp. 1377-1382). IEEE.

[20] Beigbeder T, Coughlan R, Lusher C, Plunkett J, Agu E, Claypool M. *The effects of loss and latency on user performance in unreal tournament 2003Âő.* In Proceedings of 3rd ACM SIGCOMM workshop on Network and system support for games 2004 Aug 30 (pp. 144-151). ACM.

[21] Bouch A, Kuchinsky A, Bhatti N. *Quality is in the eye of the beholder: meeting users' requirements for Internet quality of service.* In Proceedings of the SIGCHI conference on Human Factors in Computing Systems 2000 Apr 1 (pp. 297-304). ACM.

[22] Sheldon N, Girard E, Borg S, Claypool M, Agu E. *The effect of latency on user performance in Warcraft III.* In Proceedings of the 2nd workshop on Network and system support for games 2003 May 22 (pp. 3-14). ACM.

[23] Yamamoto L, Beerends JG. *Impact of network performance parameters on the end-to-end perceived speech quality.* In Proceedings of EXPERT ATM Traffic Symposium 1997 Sep.

[24] Bhandari A, Sangal AL, Kumar K. *Performance metrics for defense framework against distributed denial of service attacks.* International Journal on Network Security. 2014 Apr 1;5(2):38.

[25] Shah H, Shah P, Naik S. *DDOS Protection by Dividing and Limiting.* International Journal of Computer Applications. 2016;155(11).

[26] Ronen E, Shamir A, Weingarten AO, OÂĂŹFlynn C. *IoT goes nuclear: Creating a ZigBee chain reaction.* In 2017 IEEE Symposium on Security and Privacy (SP) 2017 May 22 (pp. 195-212). IEEE.

[27] MMD. [n. d.]. *Malware Must Die - Mirai Malware.* http://blog.malwaremustdie.org. [Accessed 21-05-19].

[28] Antonakakis M, April T, Bailey M, Bernhard M, Bursztein E, Cochran J, Durumeric Z, Halderman JA, Invernizzi L, Kallitsis M, Kumar D. *Understanding the mirai botnet.* In 26th USENIX Security Symposium (USENIX Security 17) 2017 (pp. 1093-1110).

[29] DLink Camera. [n. d.]. *DCS-942L.* http://www.dlink.com. [Accessed 21-05-19].

[30] Hackers Online Club. [n. d.]. *Phlashing-PDoS.* http://hackersonlineclub.com. [Accessed 21-05-19].

[31] Arstechnica. [n. d.]. *BrickerBot-Permanent Denial of Service.* https://arstechnica.com. [Accessed 21-05-19].

[32] ZDNET. [n. d.]. *Bricker Bot Malware.* http://www.zdnet.com. [Accessed 22-05-19].

[33] Jhaveri RH, Patel SJ, Jinwala DC. *DoS attacks in mobile ad hoc networks: A survey.* In 2012 second international conference on advanced computing communication technologies 2012 Jan 7 (pp. 535-541). IEEE.

[34] Jawandhiya PM, Ghonge MM, Ali MS, Deshpande JS. *A survey of mobile ad hoc network attacks.* International Journal of Engineering Science and Technology. 2010 Sep;2(9):4063-71.

[35] Kannhavong B, Nakayama H, Nemoto Y, Kato N, Jamalipour A. *A survey of routing attacks in mobile ad hoc networks.* IEEE Wireless communications. 2007 Oct;14(5):85-91.

[36] Zargar ST, Joshi J, Tipper D. *A survey of defense mechanisms against distributed denial of service (DDoS) flooding attacks.* IEEE communications surveys tutorials. 2013 Nov;15(4):2046-69.

[37] Mirkovic J, Prier G, Reiher P. *Attacking DDoS at the source.* In 10th IEEE International Conference on Network Protocols, 2002. Proceedings. 2002 Nov 12 (pp. 312-321). IEEE.

[38] Mirkovic J, Dietrich S, Dittrich D, Reiher P. *Internet denial of service: attack and defense mechanisms* (Radia Perlman Computer Networking and Security).

[39] Mirkovic J, Arikan E, Wei S, Thomas R, Fahmy S, Reiher P. *Benchmarks for DDoS defense evaluation.* In MILCOM 2006-2006 IEEE Military Communications conference 2006 Oct 23 (pp. 1-10). IEEE.

[40] Mirkovic J, Reiher P, Fahmy S, Thomas R, Hussain A, Schwab S, Ko C. *Measuring denial of service.* In Proceedings of the 2nd ACM workshop on Quality of

protection 2006 Oct 30 (pp. 53-58). ACM.

[41] Mirkovic J, Hussain A, Wilson B, Fahmy S, Reiher P, Thomas R, Yao WM, Schwab S. *A user-centric metric for denial-of-service measurement.* In Proc. of Workshop on Experimental Comp. Sci 2007 Jun 13.

[42] Athanasopoulos E, Anagnostakis KG, Markatos EP. *Misusing unstructured p2p systems to perform dos attacks: The network that never forgets.* InInternational Conference on Applied Cryptography and Network Security 2006 Jun 6 (pp. 130-145). Springer, Berlin, Heidelberg.

[43] Naoumov N, Ross K. *Exploiting p2p systems for ddos attacks.* InProceedings of the 1st international conference on Scalable information systems 2006 May 30 (p. 47). ACM.

[44] Qi M. *P2P network-targeted DDoS attacks.* In2009 Second International Conference on the Applications of Digital Information and Web Technologies 2009 Aug 4 (pp. 843-845). IEEE.

[45] PerakoviÄĞ D, PeriÅąa M, CvitiÄĞ I. *Analysis of the IoT impact on volume of DDoS attacks.* In33rd Symposium on New Technologies in Postal and Telecommunication Traffic (PosTel 2015) 2015 Jan 1 (pp. 295-304).

[46] Kolias C, Kambourakis G, Stavrou A, Voas J. *DDoS in the IoT: Mirai and other botnets.* Computer. 2017;50(7):80-4.

[47] Nagpal B, Sharma P, Chauhan N, Panesar A. *DDoS tools: Classification, analysis and comparison.* In2015 2nd International Conference on Computing for Sustainable Global Development (INDIACom) 2015 Mar 11 (pp. 342-346). IEEE.

[48] Helalat SM. *An Investigation of the Impact of the Slow HTTP DOS and DDOS attacks on the Cloud environment.*

[49] Farina P, Cambiaso E, Papaleo G, Aiello M. *Understanding ddos attacks from mobile devices.* In2015 3rd International Conference on Future Internet of Things and Cloud 2015 Aug 24 (pp. 614-619). IEEE.