A FRAMEWORK TO MITIGATE PROPAGATION OF IOT BASED BOTNET BY PATCHING INTERMEDIARY NODES

By

Athar Muneer

A thesis submitted to the faculty of Information Security Department, Military College of Signals, National University of Sciences and Technology, Rawalpindi in partial fulfillment of the requirements for the degree of MS in Information Security

Oct 2018

ABSTRACT

Presently Internet of Things (IoT) devices perform variety of services ranging from control of home appliances to their application in industrial sector. This technology has achieved broad acceptance by all sectors of life due to its features like low cost, energy efficient, and availability. According to a report there will be 20.4 billion IoT devices by 2020 [1]. Cyber criminals use this wide spread use of IoT as an amplifying platform to launch cyber-attacks [2]. IoT devices consume low power and have less computational capability. Complex cryptographic solutions are not considered efficient or feasible in IoT world. These devices do not have very convenient user interface to facilitate complex password management and regular patching of firmware. IoT objects have become 'once fix and remain on' type of devices. Users do not bother till the device is working. In most cases IoT devices have 24/7 Internet connection. Such environment ideally suits the cyber criminals. In the absence of common industry standards and less user awareness, criminals can breed IoT based botnet without much difficulty. These botnets can be used to carryout various malicious activities including DDoS attack against a particular target. Considering the constraints of IoT realm, updating large scale IoT devices remains a challenge. The main contribution of this thesis work is a patching scheme which aims at patching intermediary nodes to mitigate the propagation of IoT botnet. This thesis work gives an overview of main building blocks of IoT technology, overview of IoT botnets, analyses a framework to secure IoT assets. Finally it presents a gateway patching scheme to mitigate the propagation of malware.

ACKNOWLEDGEMENTS

First of all I am extremely thankful to Allah Almighty for His endless blessings bestowed upon me. I am immensely grateful to my supervisor Maj (Retd) Muhammad Faisal Amjad for his worthy supervision and support that enabled me to complete my thesis work. I would also like to thank my committee members, Asst Prof. Dr. Hamad Afzal and Asst Prof Waleed Bin Shahid for their valuable technical support and worthy guidance. Further I am obliged to all my teachers and colleagues for their endless support. Finally I would thank my parents and family for their continuous help and prayers to complete this work.

TABLE OF CONTENTS

3. PROPOSED PATCHING SCHEME

4. CONCLUSION

LIST OF FIGURES

LIST OF TABLES

Chapter 1

INTRODUCTION

1.1 Introduction

DDoS attacks based on IoT botnets are rampant. Largest of such attacks (1.2 Tbps) was in Oct 2016 against Dyn Inc.(a DNS service provider) [3]. The mirai botnet was used in this attack. Mirai is a piece of malware that constantly scans Internet for vulnerable IoT devices [3]. Mirai targets the most common vulnerability, a default username and password. As Internet use became ubiquitous, various organizations ranging from government agencies to commercial competitors shifted their business on the Internet. This reliance on Internet coupled with the unprecedented growth in the IoT technology has also attracted the cyber criminals. Large number of small IoT devices if compromised and formed part of a botnet, can create huge impact with their combined power. The cyber attackers are now well-organized, efficiently connected, highly skilled, resourceful and in some cases ahead of industry. Today's attacks are huge in their effect and incredible in their conduct as they leave no tracks to trace. Though criminal, hackers are skilful thorough professionals who take hacking as a good profitable business. Threat landscape has changed. Many risk perceptions are now real hazards. Organizations are susceptible to DDoS attacks because they either do not consider it as a threat vector or have negative satisfaction over their security arrangements. Such Organizations are secure till the time they are not targeted. An Internet asset is prone to vulnerabilities and needs proper security considerations at every phase from planning, installation, configuration, operations till maintenance.

Huge numbers of heterogeneous IoT devices are difficult to update or patch due to inconvenient user interface, lack of awareness and interest. These unpatched devices pose a great threat to entire network. A compromised device scans the network and infects other devices with same vulnerabilities at a fast pace. If gateway is patched the malware cannot propagate to other subnets.

1.2 Limited Research in the Area

The source code of Mirai-an IoT based botnet was released in Oct 2016. After release of source code and the destruction capability of the botnet one would expect that industry

and academia would join heads and come out with a solution to the problem [2]. But what we see is other way round. There is a rapid growth in the variations and mutations of Mirai. *Persirai*, *Hajime*, *Brickerbot* and *Reaper* are the names of botnets that followed Mirai and have compromised millions of connected devices which had limited or no security. The unprecedented threat vector and rise in the botnet attacks demand that more research work needs to be done to stop the fast growth of botnets and propagation of IoT based malwares. IoT gateway is a main component of the IoT infrastructure. Most of the research work has focused on detection of the malware, traffic analysis and differentiating IoT traffic, normal traffic and malware traffic passing through the gateway. Hassan Habibi *et al*. [4] have analyzed the gateway traffic. Their analysis differentiates the IoT traffic from other network traffic. It also helps in identifying the end device. But no solution to stop any malicious traffic has been provided. B. Kang *et al.* [5] have proposed a smart gateway which automatically adds new devices to a home network without human intervention. But their solution is limited only to small scale networks and has no security features. There is rich research work on gateway traffic analysis and detection of unusual traffic. The work done by S.M. Cheng *et al.* [6] has opened a new direction in security of IoT networks. They have introduced a patching scheme which patches the gateways according to descending order of gateway traffic i.e. gateways with highest traffic will be patched first. However, they have considered only one hop traffic i.e. traffic between gateway and device.

1.3 Motivation

This research is inspired by the fact that response of security experts is not in commensuration with the threat landscape created by widespread use of IoT. Source code of mirai botnet was made public in Sep 2016. Since then there is an increase of IoT botnets and attacks. Patching of huge number of IoT gateways and devices is difficult, even the presence of malware is known. There is a requirement of more work on the security management of IoT devices. If these devices are not regularly patched or updated, they remain vulnerable to be exploited and use for malicious purpose without the knowledge of their real owners. Another motivation for the research is that the trend of IoT devices is on the increase in Pakistan. Every other user is carrying an IoT intermediary device in the form of a smart phone. There are four mobile operators Mobilink/Warid, Ufone, Zong and Telenor in Pakistan. They have thousands of BSs (Intermediary devices) spread all across the country. Presently free ISM radio bands are

used for general purpose IoT devices. ISM bands have certain restrictions like limited power use to transmit signals, hence have short range. Cellular IoT uses cellular network for IoT communication, hence more powerful signals and security. This research work can be applied in any large scale network which has various numbers of intermediary nodes.

1.4 Identification of the Problem Area

IoT technology has some peculiar problems like heterogeneity and no unified international standards. The solution provided by one manufacturer does not hold suitable for others. User interface of IoT devices is generally inconvenient. Therefore users do not bother to update/patch IoT devices and change passwords. These large scale unpatched IoT devices have become a threat. Therefore we see the involvement of IoT devices in almost all newly discovered botnets. There has been a lot of research work in the area of botnet detection. But botnet mitigation techniques need more focus and work. Organizations that rely on IoT technology come across many security challenges. They first have to secure their own systems and then forestall the attempts to use their devices as part of botnet by the hackers. Stopping the spread of IoT malware is an issue, even if the presence of malware is known. Patching of IoT devices is difficult due to inconvenient user interface. Therefore any IoT malware spreads at a fast pace in IoT world. There is a dire need of a comprehensive security and management framework for managing an IoT network specially a large scale industrial network. The framework should cover all aspects related to security of an IoT network with a special focus on mitigating the spread of IoT network.

1.5 Problem Statement

DDoS attacks that involve IoT devices are on the rise. One of the reason of such attacks is that IoT devices are not regularly patched due to inconvenience and huge variety of devices. Therefore Patching of large number of heterogeneous IoT devices is difficult.

1.6 Research Objectives

The research methodology used in this thesis is applied research where existing approach to solve a problem of IoT gateway patching has been analyzed and a new approach is suggested. The new approach meets the desired results. The objectives of this research work are:-

- Study the ecosystem of modern IoT botnets with emphasis on P2P botnets.
- Analysis of different detection and mitigation approaches.
- Identify the practical challenges involved in IoT botnet.
- Present a framework which patches intermediary nodes depending on volume of traffic and other parameters.

1.7 Contributions

The contributions of this thesis work are enlisted as following:-

- It gives an overview of IoT technology.
- A brief overview of IoT botnets is presented in this work.
- It presents a framework on the basis of already existing framework to mitigate the propagation of IoT malware.
- It studies a gateway patching scheme and points out practical challenges.
- It presents a gateway patching scheme which considers all important parameters of a network traffic i.e. criticality, traffic, size/load and on the basis of the values of these parameters arranges the gateways in a descending order. Gateways on top of the list are patched first.

1.8 Thesis Outline

The presented thesis is composed of four chapters:

- The first chapter presents introduction to the topic, its importance and motivation of research work. It also includes problem statement and research objectives.

- The second chapter describes main building blocks of IoT technology i.e. IoT communication models, deployment strategies and their security analysis, four stages of Industrial IoT adoption and finally it describes rise of DDoS attacks and different types of IoT botnets which is a big threat to IoT and Internet technology.

- A framework of security and management of IoT assets is proposed in third chapter followed by the analysis of a gateway patching scheme and our own proposed solution which is a patching scheme considering all important

parameters of a network.

- Chapter 4 is the last chapter which includes conclusion and directions for future work.

Chapter 2

LITERATURE REVIEW

2.1 Introduction

IoT was introduced in 1999 and it captured the big markets in 2014 [7]. Today, IoT is part of our daily life. IoT has its presence in almost everywhere. Industries are benefitting from the intelligence provided by the connected physical objects. This intelligence is used to transform their business processes and improve productivity. Heterogeneity of constrained devices, wireless media, variety of lightweight protocols, different standards, inherent security vulnerabilities make IoT a complex technology. So any organization investing in this technology must assess the user requirements, finances, security issues, QoS, media, bandwidth and scalability before procuring any solution. Without prior deliberation, organization may face critical issues right from the outset.

This chapter describes main building blocks of IoT technology i.e. IoT communication models, deployment strategies and their security analysis, four stages of Industrial IoT adoption and finally different types of IoT botnets which is a big threat to IoT and Internet technology.

2.2 IoT Communication Models

IoT technology enables presence of virtually every physical object on the Internet. Devices also communicate with each other and can send their data on the Internet for analysis and storage. Devices can transport data from the edge site over the Internet with or without intermediary gateway. Internet enables remote monitoring and control of these objects. How Internet connectivity of the physical objects should be established? It depends on the requirement. There are following four networking models for IoT [8]:

2.2.1 Device-to-Device (D2D) Connectivity

In this model two devices which can be of different manufacturers can communicate with each other without the control of a central node. These devices can use IP over Internet or can also use lightweight protocols like ZigBee, Z-wave or Bluetooth to establish their D2D communication. The example of D2D communication model is

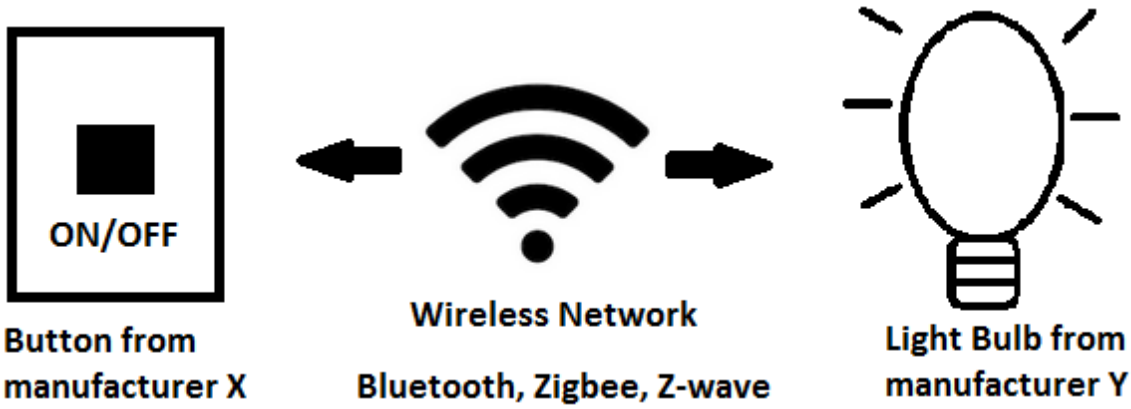


Figure 2.1: Device to Device Connectivity Model

depicted in Fig. 2.1 where two wireless devices from two different manufactures can directly communicate with each other using lightweight protocols.

2.2.1.1 Advantages of D2D Connectivity

- It reduces the traffic load on backbone network infrastructure.
- There is no dependency on the backbone network; therefore it is best suited for an emergency service when connection to cloud service is disrupted.
- It is completely self-relied with no dependence on Cloud service.
- D2D network is easy to setup.
- Cost effective.
- Useful in home automation where range is small and devices trigger action/alarm when they detect any change in the surrounding.
- Internally controlled, therefore easy to manage.
- D2D communication can be used to provide existing services and it can also support variety of increasingly new applications.

2.2.1.2 Challenges of D2D Connectivity

- Compatibility and interoperability issues can arise due to diverse nature of IoT devices.

- Security in D2D communication is a big challenge. Firewall policies deployed in traditional network cannot be deployed in intra-network D2D communication.
- Degradation of communication due to interference by nearly communicating devices.
- Authentication and admission of new devices to the network.
- Specialized discovery mechanisms are required to discover any faulty node.
- If D2D communication is established in an unlicensed spectrum using Wi-Fi and Bluetooth, it cannot embrace the exponential growth in proximity based services and devices.

2.2.2 Device-to-Cloud Connectivity

In this model devices do not need high processing or power. They have small size firmware with some rules of pushing the data to a Cloud server. In this model IoT device directly connects the Internet Cloud or application service provider. Traditional wired Internet connection or Wi-Fi is used to establish connectivity with the devices. In this model IoT devices have the processing power to use IP protocol stack. In this model gateways establish connection between IoT devices and Internet cloud. Gateways transfer data and commands between devices and web servers. Users can interact with the device through Cloud servers rather than directly communicating with the device [9]. Different manufacturers use this model to directly communicate with their devices installed at customer's location. Such connectivity facilitates both customer and vendor to improve the QoS. In recent years there is a shift in computer technology. The focus is on providing Cloud infrastructure having centralized processing, continuous monitoring, analysis and scalable storage capabilities rather than the decentralized desktop computing [10]. Device-to-Cloud connectivity is depicted in Fig. 2.2.

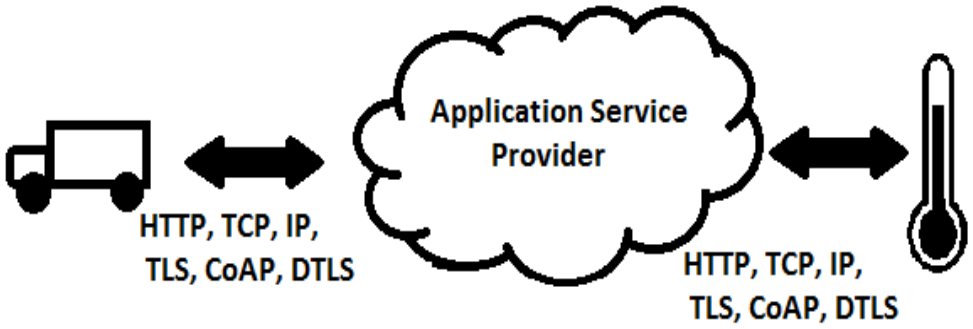


Figure 2.2: Device-to-Cloud Connectivity Model

2.2.2.1 Advantages of Device-to-Cloud Connectivity

- Centralized monitoring.
- Devices can be updated Over-the-air (OTA).
- Devices can be remotely controlled.
- Better device management i.e. admission, authentication and fault management.
- Better data management.
- Resource (power and processing) requirement for a device in this model is not much so it can help to reduce the cost.
- Control Interface resides on a cloud server which is easy to manage any change for future requirement.
- International standards can be easily implemented on centralized server.
- This model suits applications which require continuous logging for example a thermostat to monitor the temperature. This log is later used to analyze the temperature like maximum, minimum or average temperature during the day.

2.2.2.2 Challenges of Device-to-Cloud Connectivity

- Only best suited for applications where data can be pushed from the device [11].
- Inflexibility, as the decisions about data to be pushed from like which, when, and how often are made when updating the firmware.
- Internet connection is required to transmit data directly from device to a

Cloud server. If the Internet connection fails the system will not work as the system relies on the Internet.

- Data is stored on a server which is controlled by some other party so there can be data privacy issues.

2.2.3 Device-to-Gateway Connectivity

In this model IoT device connects to the Internet via gateway i.e. no direct connection with the Internet Cloud. The gateway is installed with software to communicate with IoT and perform some additional tasks like data analysis before sending to cloud and security. This connection is also called as Device to Application Layer Gateway (ALG) connection. Device-to-Gateway model can be found in many IoT devices. In some cases a costumer's mobile acts as a gateway to forward IoT device data to the Internet Cloud service. In such case IoT device manufacturers also develop a mobile app which can be used to interact with the device. In this model gateway plays an important role of interoperability in the legacy devices which cannot communicate on IPv6 or other Internet protocols. Device-to-Gateway model is depicted in fig. 2.3.

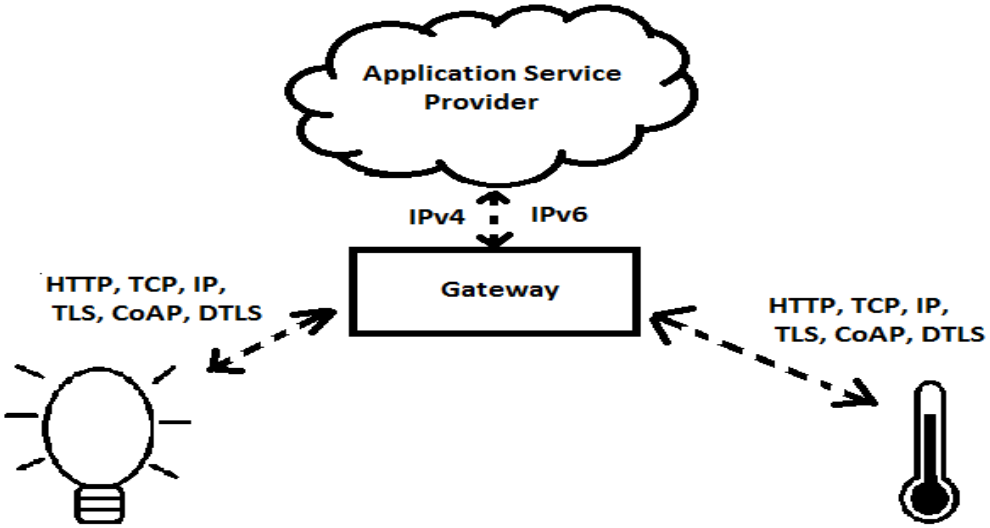


Figure 2.3: Device-to-Gateway Connectivity Model

2.2.3.1 Advantages of Device-to-Gateway Connectivity

- Additional layer of security at the smart intermediary gateway.
- Gateway can analyzed the traffic and can discard unnecessary or redundant data.
- Reduced traffic load on network.
- Reduced amount of data at the cloud.

- Monitoring of the network can be performed at local level.
- In many cases a smart phone can act as local gateway.
- Devices do not require Internet connection. The data is analyzed and aggregated at gateway. Only filtered data is transported to Cloud by the gateway when Internet connection is established. Internet connection required at the gateway.
- Gateway device can bridge the compatibility or interoperability gap between lightweight protocols like ZigBee or Z-wave and Internet Protocol (IP).

2.2.3.2 Challenges of Device-to-Gateway Connectivity

- Not Suitable for applications which require real time response like online gaming.
- An application is required at the gateway which needs to be regularly managed and updated.
- Additional cost of intermediary gateway.
- Power requirement for the intermediary gateway. Today IoT devices are available with long battery life. There are some IoT devices which can get energy from solar, heat or motion. Power requirement of gateway must be considered before deployment of Device-to-gateway connectivity model.

2.2.4 Backend Data Sharing Model

This model is similar to Device-to-Cloud communication model with an additional facility to users to export sensors' data to third parties. Users can analyze their data from a cloud service along with data from other resources. This communication model enables the data from single IoT device to be analyzed and aggregated with other data streams. For example a Smart City Management System administrator would like to have access to all important data like weather, water level in reservoirs, electricity consumption, drainage system, traffic system etc. Data from different Cloud service providers will be collected and analyzed. While in Device-to-Cloud communication model data resides with single Cloud service provider. Backend data sharing model is shown in fig. 2.4

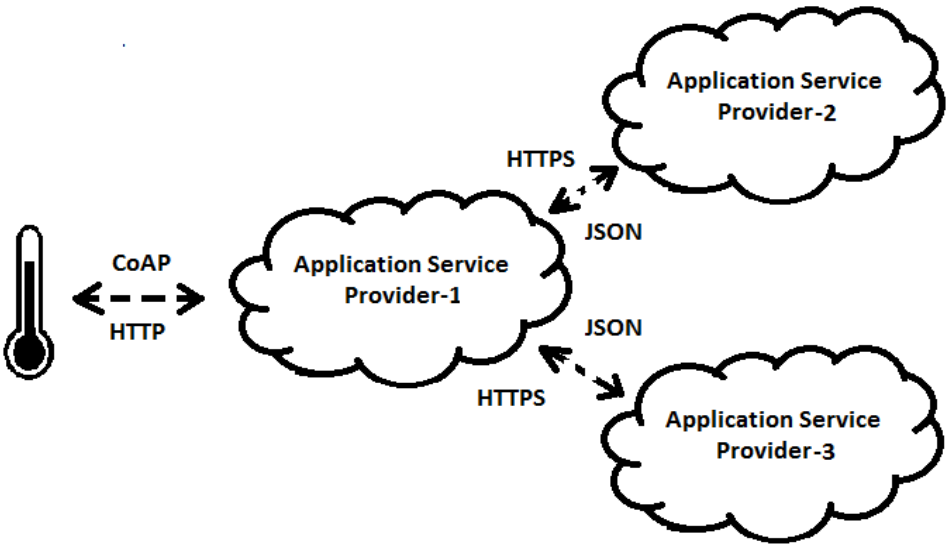


Figure 2.4: Backend Data Sharing Model

2.2.4.1 Advantages of Backend Data Sharing Model

- Better data sharing among different service providers.
- Data streams from different sources contribute to collective intelligence.
- Improvement of response and QoS.

2.2.4.2 Challenges of Backend Data Sharing Model

- Reliance on Internet to provide Cloud services.
- Big Data management.
- Data privacy and authentication.
- Tracking the data source in case of any issue.

2.3 IoT Deployment Strategies

IoT is changing our living style in a way we never thought. New IoT devices and applications are being introduced in the market at affordable prices. Many enabling technologies like wireless communication, MEMS, RFID, Cloud Computing and digital electronics have been fused to form IoT. There are different IoT deployment strategies. In this world of everything connected with the Internet, it is really important to have good deliberation for deployment of an IoT system for your home/organization. Assess your requirement, budget, available technologies, and future scalability and then decide

about a particular solution. Without such homework it will be like a walk in in a deep channel without a light. Many approaches can be used to deploy an IoT setup to achieve required services [12]. Mainly there are two approaches i.e. Centralized and Distributed approach [13].

2.3.1 Centralized Deployment Approach

Centralized approach is client-server architecture in which data aggregation and control is performed at a central unit. Edge sides "*things*" are used to sense collect and transport data. *Things* also receive instructions from the server/central unit and perform particular action. In this approach IoT technology relies heavily on cloud computing. The data from IoT devices is collected, analyzed and stored at central cloud server. The central server controls the access and sharing of huge volumes of data. Data can be provided to end user and it can also be shared among other service providers depending on the type of data, application and environment. Any new connection to the network is controlled by the central unit. Management of the network is performed at a central unit. IoT devices can be accessed through the central device. This central device can be a gateway, server or cloud service which controls all connections and inbound/outbound traffic. The central service has huge computational capacity and storage media. It can perform intensive tasks to transform IoT data into intelligence and use it for human value. Centralized deployment approach is depicted in figure 2.5 in which gateway or a smart phone acts as an intermediary device. This is like a bridge between IoT device and a cloud service.

2.3.1.1 Security Management in Centralized Deployment

Millions of IoT devices are being used these days. This number is increasing day by day. An IoT device is a tiny little device with small processing capability which works on limited battery life. If a single device is compromised, it will not have any phenomenal effect but when millions of such small devices are compromised and formed part of botnet, can create a huge impact. Therefore security of IoT system is as important as it is in any other information system.

Centralized approach provides a layer of security in IoT. Any entry of new device is allowed through a central unit. Monitoring and implementation of policies is also carried out at the central device. Central approach provides a good security management. However, Centralized approach relies on central server for enforcement of policies and

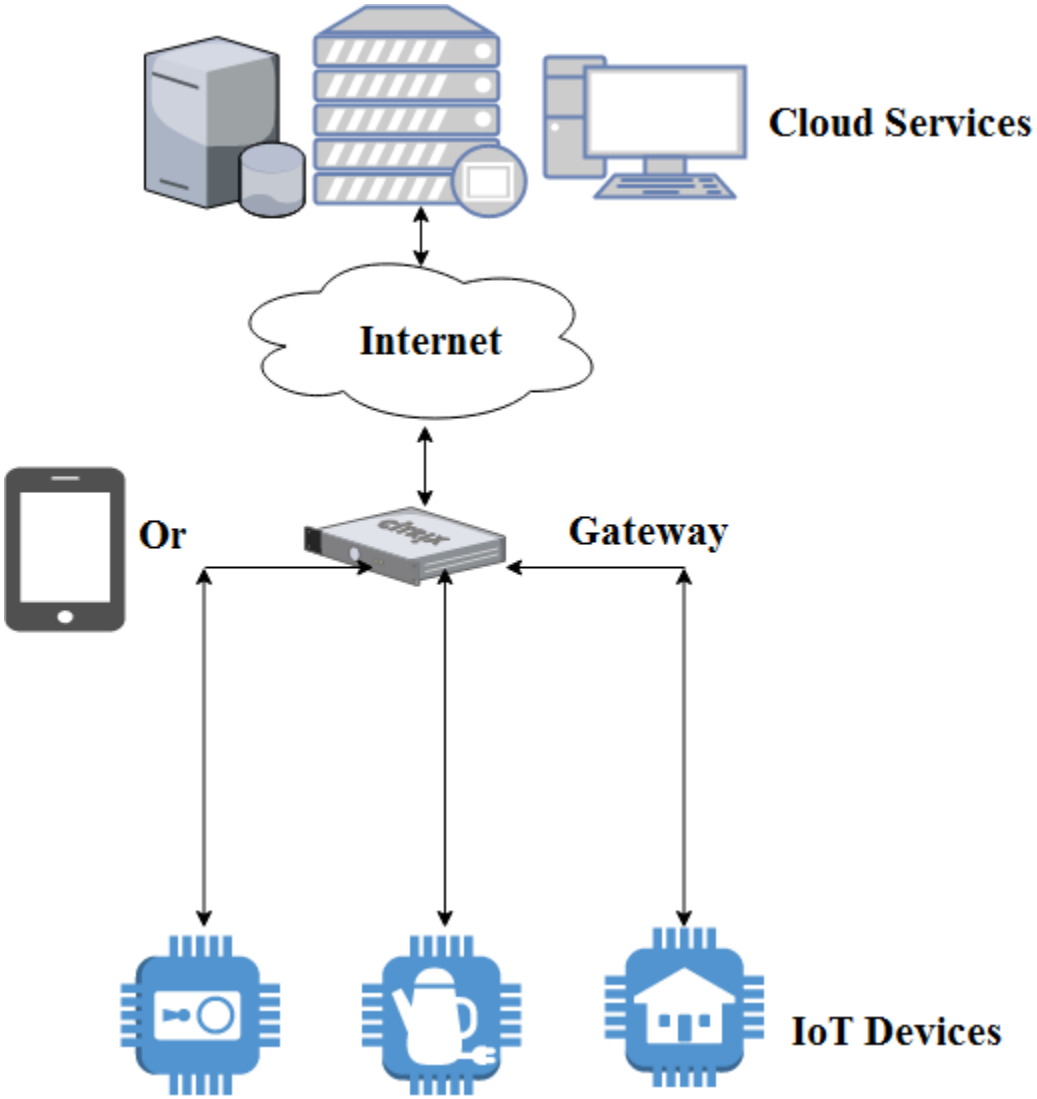


Figure 2.5: IoT Centralized Deployment Approach

security; hence it has a single point of failure. If any loophole is exploited with a botnet of millions of devices, the whole system can be taken down.

2.3.1.2 Advantages of Centralized Approach

Centralized approach is being used in many IoT applications. Some of the advantages of centralized approach are given as under:-

- Good access control and better management of resources by system administrators.
- Strong security measures like IPS/IDS, encryption and antivirus can be deployed at central device/cloud service.
- Updates and patches can be easily installed through central device.
- Data sharing, privacy and tracking can be achieved with high computational

capability at a central device.

2.3.1.3 Disadvantages of Centralized Approach

Central approach provides a good control, management and monitoring but it has certain disadvantages which are given as under:

- Central approach provides a single point of failure. Therefore it is prime target for criminals.
- If central device is compromised then attacker can do hell of things i.e. steal data, jam the network, redirect traffic, corrupt/ destroy the data and launch attack against another target.
- Any unintentional misconfiguration can jeopardize the system leading to downtime or self-inflicted DoS attack.

2.3.2 Decentralized or Distributed Deployment Approach

Centralized approach is good for management. But there are massive numbers of connected devices around the globe. This number is increasing at a fast pace. Due to this unprecedented increase in the number of connected devices the central devices/servers can suffer from shortage of resources and computational capabilities. Therefore a decentralized approach is used where tasks of the central devices like processing and storage can be offloaded to edge side. In a decentralized approach devices collaborate with other devices and work autonomously. System will keep on working if one of the collaborating devices fails. Diagram of Decentralized Approach is given in figure. 2.6.

2.3.2.1 Security Management in Decentralized Approach

The centralized approach has main vulnerability of single point of failure. To reduce the security vulnerabilities in centralized approach more and more processing has to be shifted to the edge. Decentralized security although difficult to manage, removes inherent threats attached with the centralized approach. To meet the requirements of scalability, decisions and processing of data needs to be done locally. Each IoT device can have different processing capability and power duration. In decentralized approach each device has to be configured and secured separately. Security management is difficult in a decentralized approach. However, it is very difficult to break decentralized system. The adversary will have to compromise all nodes in order to completely jeopardize the system.

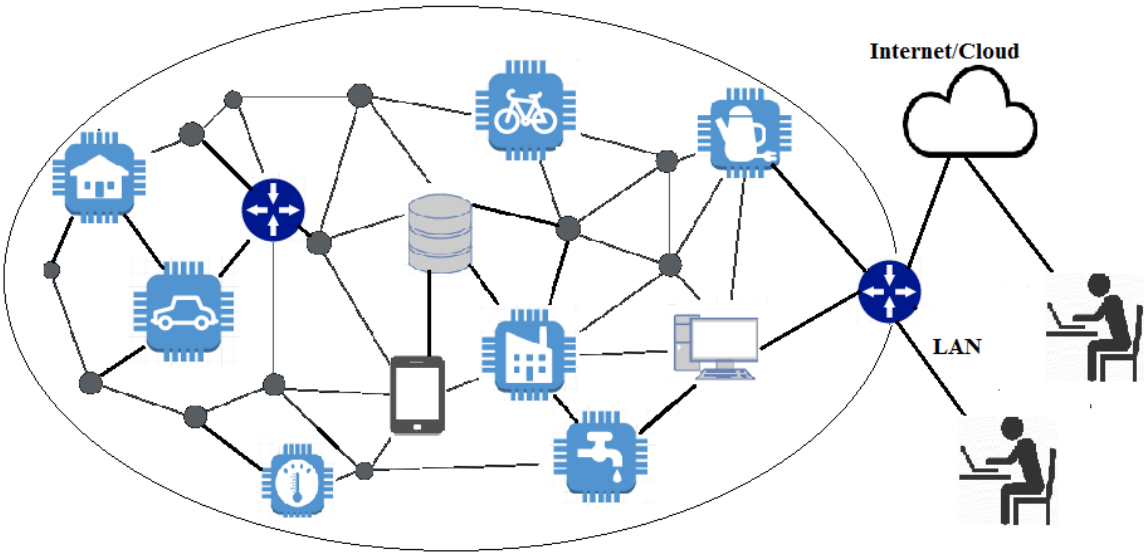


Figure 2.6: IoT Decentralized Deployment Approach

To achieve this, the adversary should have enormous computing capability which is not possible in normal circumstances.

2.3.2.2 Advantages of Decentralized Approach

In a decentralized approach devices communicate with each other for data sharing and other services. Advantages of decentralized approach are given as under:

- New devices can be added to the network dynamically as there is no centralized control. Technologies like proximity based device authentication are helpful to authenticate new devices.
- If one device goes down, others will keep on working as prescribed. There is no single point of failure.
- This approach suits large networks. There is no pressure on network bandwidth or central devices.

2.3.2.3 Disadvantages of Decentralized Approach

- Difficulty in managing large number of devices as there is no central control.
- Patch management and updates without a central control are difficult to manage.
- Implementation of policies across all devices is difficult to achieve.
- Consistency of data is difficult to manage.

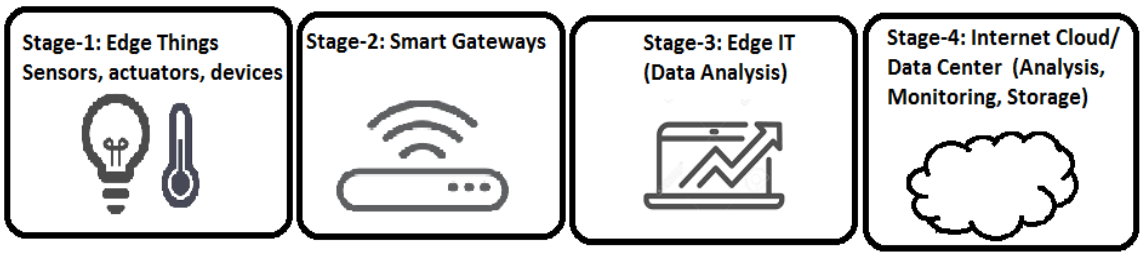


Figure. 2.7: IIoT Adoption: Four Stage Architecture

2.4 IoT Adoption: 4 Stage Architecture

IoT is not just a network of wired or wireless sensors; it is a technology which integrates various processes like sensing physical environment, processing at edge site, forwarding the data to Cloud/Data Centers for analysis, decisions and storage. There are two main categories of IoT application i.e. IoT for consumers and Industrial IoT (IIoT). In first category consumers use wearable IoT smart devices or use IoT based smart home network to run, monitor and control home appliances like air-conditioners, refrigerators, microwave oven and lighting. In IIoT devices are used to improve productivity and life security. IIoT devices need to operate in rugged environments. These devices are very critical and robust. In this section focus is on IIoT. IoT adoption at industrial level has four stage processes [14] depicted in Figure 2.7:-

2.4.1 Stage-1: Edge Things

Sensors and actuators act as eyes/ears and hands of IoT architecture. Sensors are the actual source of data. In industrial applications of IoT, sensors can be used to sense different types of environmental changes like temperature, motion, pressure, humidity/ moisture, water flow, light etc. Sensors work like a transducer to convert some physical change into electrical impulses which can then be converted into digital signals or to determine the reading of change [9]. Actuators work in reverse order of a sensor to convert electrical input into physical action. Sensors and actuators are seamlessly combined into IoT nodes or *things*. Sensors collect information from the surroundings and actuators initiate some action (controlling) [15]. For example sensor will sense the water level in a container and when it reaches certain lower limit a signal will be passed to actuator which will switch on the water pump. Sensors and actuators are blend of innovation in different technologies like wireless communications, digital electronics and

micro-electro-mechanical systems (MEMS) [15]. IoT devices are resource constrained. In many cases IoT devices have very limited processing capability. However, some processing can be done on IoT data at every stage of IoT architecture [16]. More processing is done on the edge side if real time action on data is required like in case of a smart car which can immediately slow down or shut the heating engine. Processing is performed on the Cloud if deep insight and analysis on IoT data is required. It depends on the environment.

2.4.2 Stage-2 Smart Gateways

The data from sensors comes in analog form. The Data Acquisition System (DAS) converts analog data into digital format. The data is aggregated and then routed to Internet cloud or stage-3 for further processing. The main responsibility of smart gateway is to enable communication between physical devices and Cloud/data center. Sensors generate huge volumes of continuous data. This voluminous data is not always required at Cloud level. So at this stage some preprocessing is performed on the data. Once data is received a smart gateway can perform some function like discard, transform or aggregate the received data before dispatch [17]. As a result of this pre-processing some commands are sent to IoT devices. Smart gateways have the capability to understand field protocols like Bluetooth Low-Energy (BLE), ZigBee,Wi-Fi, Near Field Communication (NFC) and can convert them to Internet/Cloud protocols like Message Queue Telemetry Transport (MQTT), Advanced Message Queuing Protocol (AMQP), Constrained Application Protocol (CoAP) and HTTP.

2.4.3 Stage-3 Edge IT

After stage-2 data is aggregated and digitized and is ready to be forwarded to Cloud. But sending huge amount of data directly to Cloud can consume network bandwidth and create latency. It can also create storage and security issues at the Cloud level. Therefore some analysis is performed at the edge location which is generally close to field sensors and only meaningful or anomalous data is dispatched to Cloud for further analysis/processing. For example sensors at some water reservoir generate continuous data of water flow/level every second. It will be appropriate to analyze this data at Edge side and forward aggregated data after every 2 minutes.

2.4.4 Stage-4 Internet Cloud/Data Center

IoT technology generates huge amount of data so data analysis, management, storage and expiration needs to be dealt with carefully to get true benefit from the technology. There is a requirement of huge processing, energy and manpower for data analysis and management of voluminous data. Such processing and energy requirements cannot be completed at edge side. Internet Cloud/Data Centers perform central analysis and archiving of IoT data. In some cases stage-II and stage-III are bypassed and data is directly transferred to cloud. But with enhanced processing capability at edge side this 4-stage architecture is better. Large enterprises deploy their own data centers while many hire the Internet cloud services for archiving their IoT data. Few Internet cloud service providers are Google Cloud, Microsoft Azure, Oracle Cloud and Amazon Web Services(AWS).

2.5 Characteristics of IoT

IoT can be referred to as an umbrella term which covers the amalgamation of advancements in several technologies like wireless communication (WSN), digital electronics, embedded platforms, small lightweight OSs and protocols. IoT aims at provision of Internet connection to virtually every physical object. Today the presence of IoT in our routine life can be felt. IoT can be found everywhere from our smart homes to industries, transport, agriculture, critical infrastructures and health care systems. This huge presence of IoT devices generates volumes of data. This massive data needs to be properly managed, analyzed and stored. There are certain issues related to the ownership, control and access to IoT data. IoT relies on Cloud Computing and Big Data technologies to achieve desired results with respect to data management and control. The gains of IoT technology do not lie in provision of Internet connection to physical objects nor does it depend on the capabilities of embedded systems or sensors and actuators. The real strength of IoT is in how to take insights from IoT data and use this inference to transform business processes and models. The lifecycle of IoT data collection and its transformation into valuable information is depicted in the following fig 2.7:-

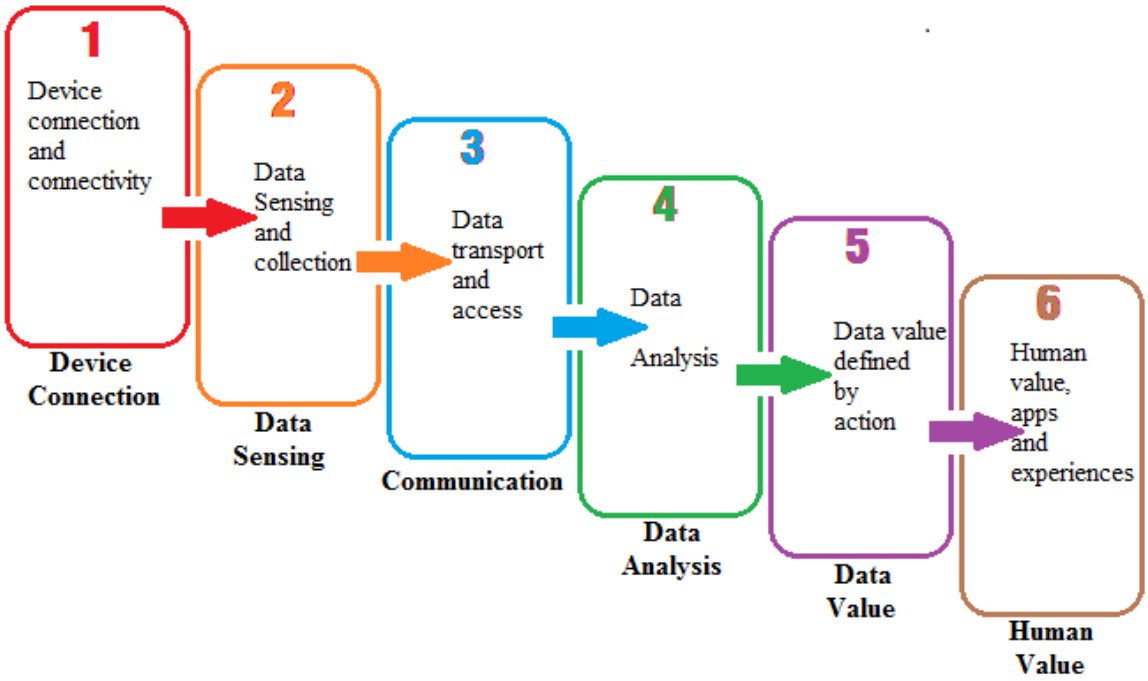


Figure. 2.8: IoT: From Connecting Devices to Human Value [18]

IoT is a vast area of study. Its characteristics can change from one environment to another [19]. But in general IoT devices have following characteristics:

2.5.1 Connectivity

IoT devices are heterogeneous and these devices can be applied in diverse fields. In an IoT environment devices can communicate with each other and can transfer data or receive commands from Internet Cloud. Therefore connectivity of an IoT device is very important. For any IoT setup following connectivity qualities are desired:

2.5.1.1 Flexibility

Connectivity should be flexible and can be provided to a device on the move. It can accommodate diverse devices which operate on different protocols.

2.5.1.2 Reliability

It should be reliable. In some applications like healthcare and critical systems 24/7 reliable connectivity is required.

2.5.1.3 Out-of-the-Box Solutions

There should be out-of-the-box connectivity solutions. Users' demand and industry requirements are ever increasing. As more and more devices are being added to the network

therefore more pressure on network bandwidth can be anticipated. In such scenario *hyper connectivity* solution is more suitable where network bandwidth and capacity is always more than the users' demand.

2.5.1.4 Quality-of-Service (QoS)

In some applications of IoT degraded or substandard service is unacceptable. QoS should be as envisioned and agreed upon between user and service provider. Since IoT devices are being deployed in very rugged environment. There can a disruption of connectivity due to various reasons like natural disasters, outbreak of fire or terrorism activity. In such scenario the service provider should have a trained staff which should work hand in glove with the user establishment to timely restore the services.

2.5.2 Intelligence

An IoT device can have some of very simple to complex algorithms, it can perform computations, it has software and hardware which make it intelligent [19]. Ambience intelligence is a quality of IoT devices that can make them perform certain actions according to a changing environment or physical condition. This individual intelligence of a device can contribute towards collective intelligence, compatibility and accessibility.

2.5.3 Dynamic Nature

The basic job of an IoT device is to collect data from the environment. This can happen when the condition or environment around the device changes. This change can be detected or sensed by the device and can be transferred in the form of data. During its lifetime an IoT devices changes its state dynamically from active to sleep mode or to save energy mode. Similarly from connected to disconnected depending on the context or change of environment around it.

2.5.4 Vast Scale

Number of IoT devices that will interact with each other and will communicate on the Internet will keep on increasing. According to a Gartner report there will be 20.4 IoT devices by 2020 [1]. There is an unprecedented growth in this sector. It will be challenging to manage such huge number of connected devices. Both performance and security will be equally important in future.

2.5.5 Sensing

Sensors are pivotal to IoT. They are like eyes, ears and nose to a human body. They detect change in the environment in which they operate and report that change. Sensors inputs are in raw form but when they are analyzed and put into perspective they can help in better understanding of the world.

2.5.6 Heterogeneity

Heterogeneity is one of the main characteristics of IoT. Devices can be as diverse as the life around us. The reason for this diversity is that they have to operate in different environments. The IoT technology should support devices from different vendors which have different energy requirement and operate on different protocols. Compatibility and interoperability in IoT world is the area which is considered highly important.

2.6 IoT Security

IoT devices have large domain of application. They range from wearable devices, smart home appliances to sensors in industrial and Critical Infrastructures (CI). Many devices share common data. This shared data has a huge amount of private information [19]. Therefore to secure this private information from unauthorized access is really challenging. Security is very critical in an IoT structure. This important aspect of technology is somehow neglected by the users and manufacturer. Manufacturers focus on creating something new that can meet the user demands and grab the maximum market share. Users want something novel with great performance at a lower cost. In this context security is difficult to fit in. Security causes inconvenience to users, requires more processing and energy hence the cost of device goes up if it comes with high standard security parameters. Therefore IoT devices are generally more vulnerable to security threats as compared to traditional desktop computers. There can be many ways a hackers can attack IoT network. IoT Infrastructure can be divided into three main parts from security perspective. These three parts which are main targets of the hacker are device, network and Cloud. The IoT security management is depicted in figure 2.8:-

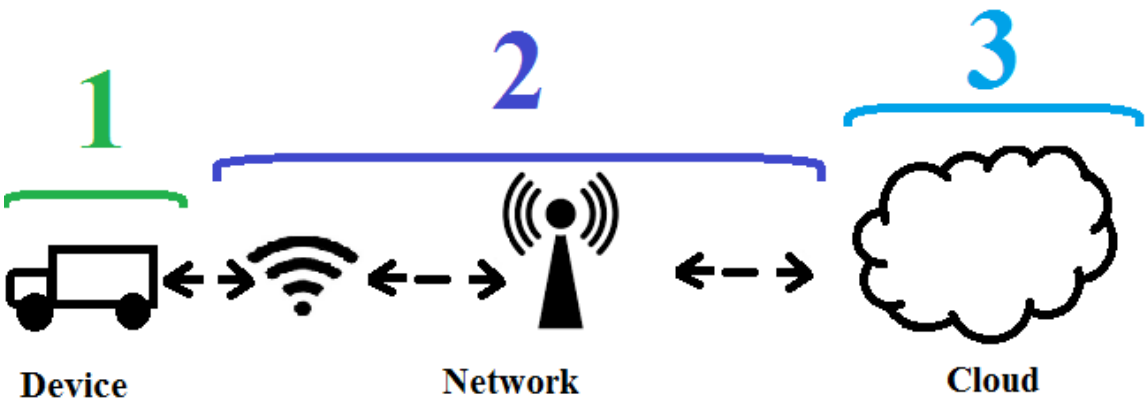


Figure. 2.9: IoT Security Management

2.6.1 Device Security

IoT devices interact with each other. They sense, collect and transfer data over the Internet or to a local data center. Therefore security of IoT devices is very important. Unfortunately IoT devices are most vulnerable part in the security management of an IoT network. Most IoT devices are compromised due to weak or default username/password. The two important IoT security measures that a user should do are to change default password and update device firmware. Similarly there are various web services and mobile apps that claim to facilitate the users interacting with IoT devices. Users should take extra care to use such apps and must check the authenticity. If there is no or weak physical security then it is also possible that somebody can tamper the device hardware. IoT device should not connect to any open Wi-Fi connection. It should be configured to only connect to your own private Wi-Fi. Security of an IoT device is also dependent on its processing capacity and battery life. A device with more processing and battery life can implement strong authentication mechanisms and encryption algorithms. Device firmware should be regularly updated. This is an area which is very difficult to ensure. Some devices do not allow over the air update firmware due to hardware limitations. Others do not have an easy user interface. Therefore a lot of IoT devices remain unpatched and hence create a huge attack surface. We have proposed a solution in this thesis which aims to stop the spread of IoT malware by patching IoT gateways.

2.6.2 Network Security

Most IoT devices operate in wireless network. They are also resource constrained devices. Therefore they suffer from inherent issues of wireless network like interference,

signal loss and other security issues. Non availability of proper monitoring system makes it more vulnerable. Attacks like message forgery, eavesdropping and changing destination of packets are common in a wireless network. Following 9itwo important factors of IoT network make its security unique and challenging:-

- Heterogeneity of large scale devices (endpoints) which operate in weak or almost no physical security.
- Aggregation of huge amount of IoT data.

Above points create a huge impact if security is compromised. The core network i.e. the network beyond IoT gateway is similar to the architecture of conventional networks. The conventional network is mature and provides security against Man-in-the-Middle (MITM) attack, impersonation, compromising confidentiality or replay attacks. The difference between conventional network and IoT lies in the edge network and type of data/traffic. The IoT traffic pattern is different from the traditional network. For example network traffic generated by humans has peculiar pattern (low, high and peak) with respect to time. Whereas traffic generated by IoT devices deployed in home, industry, agriculture, healthcare systems can have different traffic pattern. Therefore it is generally practiced to deploy a separate firewall and monitoring system for your IoT network.

2.6.3 Cloud Security

In a cloud based IoT structure huge data is collected, aggregated, analyzed, stored and provided to authorize users by the Cloud service provider. The cloud service provider has an important role in providing safe end to end communication and responsible for data security. Two main aspects are securing the privacy and location in case of mobile IoT [20]. The collected IoT data has to be shared among different authorized parties for example the data of working street lights in a smart city can be of great interest to an emergency service provider or rescue workers, police, traffic etc. The real challenge in this aggregated area is to secure the privacy of a user from public. Nobody wants that data about his routine life pattern, eating habits or arrival of guests be made public. Similarly leaking the information about a person mobility or location can be embarrassing and sometimes it can become a disaster. Privacy to healthcare data is regarded as an absolute right of a patient in some countries. So a cloud service not only secures privacy it also can track a user or the originator device in case of a dispute.

2.7 Rise of DDoS Attacks

Distributed Denial of Service (DDoS) is a kind of Denial of Service (DoS) attack in which large numbers of compromised devices forming part of a botnet are used to attack a system so that legitimate users could not have access to services. This type of attack is rampant with the ubiquitous use of IoT devices. Any system which has an Internet connection is prone to vulnerabilities and needs proper security considerations starting from planning, installation, configuration, operations and maintenance. The Cyber attackers take full advantage of the impact of IoT technology by using a botnet of compromised devices in a DDoS attack against a particular target. Following table shows the largest DDoS attacks from 2013 to 2016. Largest of the attacks was in 2016 against Dyn which involved an IoT based botnet 'Mirai':-

Table 2.1 Largest DDoS Attacks from 2013 to 2016

Month & Year	**Target & affect**
Mar 2013	Spamhaus, a European volunteer spam-fighting organization. DDoS attack using DNS amplification with a botnet. The attack slowed down global internet speeds. Attack peak traffic: 300 Gbps [21]
Nov 2014	Hong Kong's news websites Apple Daily and PopVote. Botnet involved in the reflection attack. Largest DDoS attack in the history Attack peak traffic: 500 Gbps [22]
Dec 2015	BBC website. DDOS attack. BBC website went down for multiple hours. Attack peak traffic:602 Gbps [23]
Oct 2016	Dyn Inc. (DNS service provider).Web sites including Twitter, CNN, the Guardian and many others were down for most of the day. Mirai botnet involving Internet of Things (IoT) was used. Attack strength of 1.2 Tbps [3]

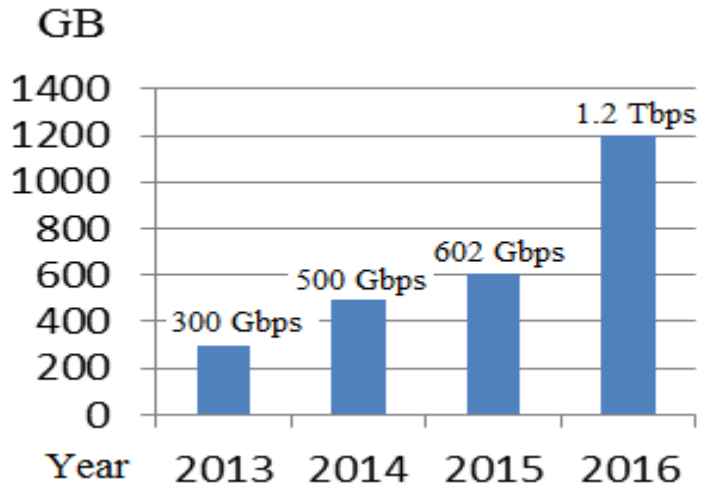


Figure. 2.10: Largest DDoS Attacks from 2013 to 2016 – Graphical Representation

2.8 IoT Botnets

IoT based botnets have become a major threat to the Internet. A well-coordinated IoT based botnet DDoS attack can take any target down in minutes. Botnet is a network of infected devices under the control of a "bot master". Botnets are used for malicious purpose on the Internet. Due to their large scale impact and weak or no security, IoT devices are prime targets of cyber criminals. Following paragraphs discuss the different topologies of botnets and why IoT botnets are preferred by the bad guys:-

2.8.1 Botnet Topologies

Traditionally Internet botnets followed client/server architecture. Bot master controlled its bot through a C&C server. But such bots could be easily detected. These days botnets use different architectures for C&C following paras discuss different topologies of botnets.

2.8.1.1 Centralized C&C

In this model all compromised devices are controlled by a central server. Bots contacted a pre-defined server to receive commands and report events to the server. All communication was performed through predefined domains or IRC chat networks. Compromised devices would continuously receive messages from a central device. Such messages have low latency [24]. But this client-server technique has two weaknesses i.e. with latest detection techniques it is possible to detect the botnet since many devices connect to C&C center at a particular time, and secondly if detected whole system can be taken down as there is only one central

device.

2.8.1.2 P2P Structure

In a P2P structure there is no central server. Each device on the network can act as a client or server. Devices can share data and receive commands from one another. P2P botnets have advantage over centralized client-server based botnets in that discovery of one bot does not necessarily lead to bot master or disruption of complete botnet. P2P botnets are difficult to detect. However, P2P botnet is complex and there is no guarantee on delivery of commands.

2.8.1.3 Unstructured Botnets

A botnet communication can be based on a system where one bot would only know about one other bot and can send and receive instructions from that bot [24]. Detection of a single bot in such a system will not compromise the entire botnet.

2.8.2 Reasons for Creating IoT Botnets

Recently there has been a surge in the IoT based botnets. Millions of devices were infected with *Bashlite* malware and its variations in 2014, *Mirai* botnet created a havoc in 2016, *Hajime*-a P2P botnet made fun of IoT security and the spread of *Persirai* botnet compromised millions of IP phone camera. If we look at the strength and trend of large DDoS attacks in previous years it is evident that every new attack is more sophisticated and powerful than the previous ones. We see variations of old botnets coming up every week. Following are some of the reasons why IoT is being considered as a soft target by the criminals:-

- IoT are considered like a low hanging fruit [25]. Most devices are found with default username/password.
- Most IoT devices remain on for 24/7. There is no issue in accessing the device during any time of the day.
- Large numbers of IoT networks have poor monitoring system. There is no fear of detection by the hackers.
- Cost of compromising IoT devices is low as compared to compromising servers in a conventional network. [25]
- Large number of compromised devices can create a huge impact.

Chapter 3

PROPOSED PATCHING SCHEME

3.1 Introduction

Currently the number of connected IoT devices is around 8 billion [26]. But unfortunately the security of IoT technology is somehow neglected. The main reason for this neglecting is that manufacturers want to get more and more profit. They want to build new devices with less cost and a quick launch in the market. If they add security to these devices, it will require more time to build and cost of the device will go up. The second main reason for weak security in IoT sector is lack of user awareness. Users think that a tiny IoT device does not contain any data so why to worry about its security. Users do not change default passwords of the devices. Fact of the matter is that IoT security is a complex and big issue. A 2016 Wind River System white paper [27] says that security of IoT systems is more challenging than conventional cyber security.

In this chapter NIST framework for improving the cyber security of Critical Infrastructures [28] has been discussed. It is suggested that same framework can also be used for IoT network. Maximum portion of this chapter is focused on *Respond* phase of the framework which highlights actions after detection of the malware. We have also analyzed an already proposed patching scheme which gives priority to gateways with high network traffic. We have suggested our own patching scheme which takes all important parameters like criticality and size along with network traffic into account and shows desired result.

3.2 NIST Cyber Security Framework for Critical Infrastructures

In 2014 NIST introduced a framework to guide private sector organizations to manage their information assets and Cybersecurity risks to those assets. It provides standards and guidelines to help organizations to improve their security stature. It is a very flexible and cost-effective process to reduce risks to information assets of an organization. The *framework core* consists of five simultaneous and continuous functions. IoT are now part of major industries and their security is also very important, as in the case of any other information asset of the organization. IoT assets and risks to those assets need to be identified and evaluated. Required risk mitigation techniques may also be applied for IoT

network. All such activities should be performed according to a well chalked out plan or framework. The purpose of any risk management framework or strategy is to protect the assets of an organization so that the organization should successfully continue its normal operations. [27] Stated that the NIST Cybersecurity Framework for Critical Infrastructure can also be applied to IoT. Five concurrent functions of the *framework core* are given as under. The activities of each function are recomposed as per the IoT environment:-

3.2.1 Identify

This function lists activities to develop the organization's understanding to deal with the risks to information assets i.e. data, capabilities, systems etc. The activities of this function are given as following:-

- Total number of assets.
- Asset owners.
- Location of assets.
- Environment in which assets are used.
- Group them in critical/non critical.
- Assign them value of criticality (e.g. 1 to 10, 10 being the most critical).
- Level of threats to the assets.
- When was last updated.
- Documentation.

3.2.2 Protect

Actual physical or technical controls are applied in this function to ensure that services of critical infrastructures continue uninterrupted. Activities of this function are given as following:-

- Procedures for physical, technical and cyber security.
- User training.
- Access control and passwords.
- Availability and testing of patches.
- Maintenance.
- Logging and documentation.

3.2.3 Detect

Develop a system and activities to monitor and identify the occurrence of any cybersecurity event. The monitoring staff should be trained to identify any breach or attempt to break the system. They should have contacts of all system/ network administrators. The activities of this functions are given as under:-

- Continuous monitoring.
- IDS or Network monitoring tool.
- Honeypots.
- Look for online vulnerabilities and patches.
- Unusual events should be identified logged and response plans should be updated.
- Communication of information.
- Reporting of events to concerned administrators.

3.2.4 Respond

Appropriate actions are taken after a cybersecurity event has occurred. Plans and activities are prepared in advance. These plans are regularly updated in the light of lessons learned. Activities in this function are listed below:-

- Response plans (during and after the incident).
- Patch the nodes according to a patching scheme.
- Sharing of information with asset owner and all stake holders.
- Containment and isolation of the breach.
- Elimination/removal of malware/ fault.
- Analysis of impact to business.
- Update response plans(document).

3.2.5 Recovery

Plans should be developed to restore the system back to its normal operations after a cybersecurity incident has occurred. In case of an event of serious nature when there is a chance of compromising data or system integrity, then system is restore to a back time or data from available backups. In case of natural disaster or damage to the infrastructure the system resotres its operations from a Disaster Recovery(DR) site. Everybody should

know its role in the recovery phase and DR and BCP should be well rehearsed. Activities in this phase are given as under:-

- Execution of recovery plan.
- Restoring the problematic node or installing new one.
- Restoration of business.
- Information sharing.
- Reputaion/public relations.
- Lessons learnt.
- Update recovery plans.
- Documentation.

3.3 IoT Gateway Patching Scheme

There have been paradigm shifts in the field of IoT. Many companies provided centralized cloud/ based solutions where intensive processing, decisions and storage activities are performed on the server side. Whereas edge side performs less processing intensive tasks i.e. sensing and transfer data to the cloud service. In such an approach gateway is just a bridge between sensors and Cloud with no intelligence or decision making.. The centralized approach has better control and management but it has single point of failure. The new shift in IoT paradigm is to infuse edge side i.e. sensors, gateways, local servers with some intelligence and offload the pressure from central cloud servers. Otherwise central servers with single point of failure will not be able to sustain the burden which is a result of rapid growth in IoT sector.

The idea of gateway patching is based on the fact that huge number of devices are left unpatched due to following problems:-

- Heterogeneous IoT devices need different patches from different vendors. Therefore it is difficult to patch all devices.
- IoT devices do not have a convenient user interface. Therefore users do not take pain or inconvenience to patch the devices.
- Procurement of large scale patches for all devices regularly can be costly

In spite of above mentioned problems, propagation of malware can be mitigated if IoT gateways are patched. A smart gateway with some intelligence and decision making capability can play an important role to secure an IoT network. S.M Cheng *et al.* [6] have leveraged the gateway capabilities and suggested a traffic-aware patching scheme

which patches the IoT gateways. The scheme generates a list of gateways in descending order of volume of network traffic. Since patches can be limited so gateways on top of the list are patched first. Our analysis of the said scheme is given as following:-

- Network traffic is an important parameter but not the most critical one. There are some other important parameters such as criticality and size (no of devices connected with the gateway).
- Volume of network traffic can be misleading in case of a diversionary attack/DoS.
- Network traffic should be dependent on size(number of connected devices) of the network. A real candidate for patching is the gateway which is generating high traffic with less number of devices. Although total volume of its traffic is slightly less than the gateway with more number of devices.
- In case of multiple gateways with same network traffic which gateway to be patched first? There should be some other criteria to deal with such case.

On the basis of above points we have proposed our patching scheme which is given in the following paras. This scheme is the main contribution of our research work.

3.4 Proposed Patching Scheme

We have introduced a new patching scheme which can help in following two problems:-

- When you have large number of intermediary gateways in your network and you have to update all gateways, which sequence you will follow? Which are the important/ critical gateways which need to be patched first?
- If you have limited number of patches and large number of gateways, which gateways you should patch immediately and leave others for a future update?

Our gateway patching scheme helps in organizing the gateways in descending order of the value of objective function *f(c,l)*. The gateways with higher value of objective function will be patched first. After patching the gateway malware cannot be propagated from one gateway to another gateway (Figure 3.1).

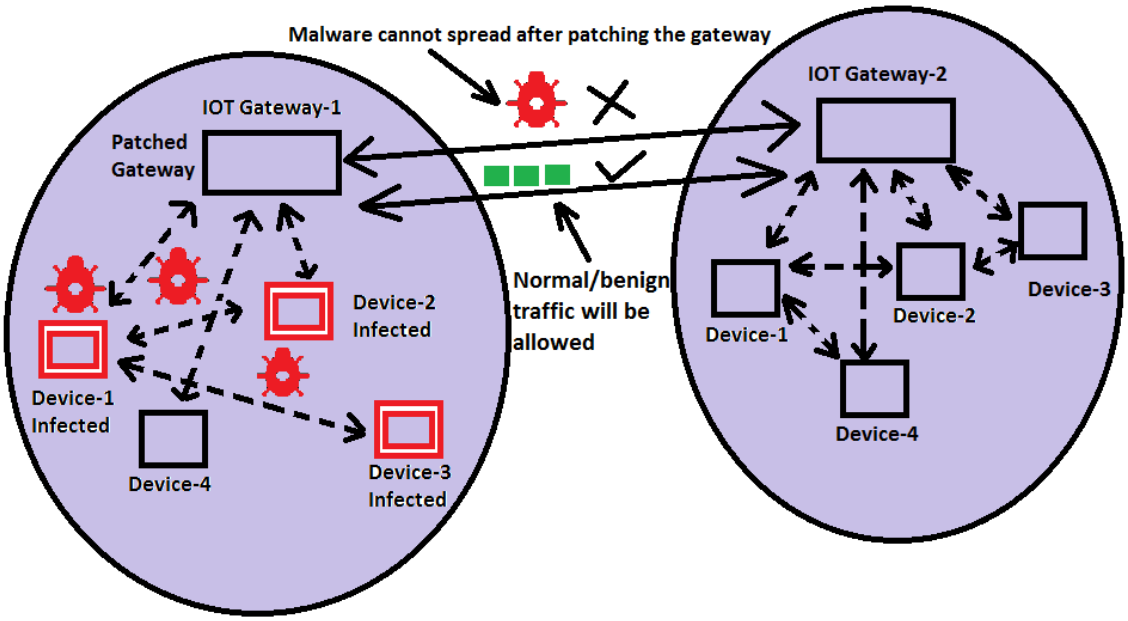


Figure. 3.1: IoT Gateway Patching Scheme

3.4.1 Features of Proposed Scheme

Proposed scheme has following features which make it different from the work done by S.M Cheng *et al* [6]:-

- We have included two important parameters i.e. criticality and network load (traffic/ size) with different weightage in our scheme.
- We consider that criticality is the most important factor w.r.t security of any network. Therefore criticality has been given maximum weightage in the proposed scheme.
- We have formulated an objective function which is based on the value of three important factors i.e. criticality, size and traffic of an IoT gateway.
- In proposed scheme value of network traffic is dependent on size (number of connected devices). Therefore only two inputs i.e. criticality and load are used in the objective function instead of three i.e. criticality, traffic and size (traffic and size are represented by load).

3.4.2 Values for Criticality, Size and Network Traffic

According to [28], all information assets need to be identified for proper risk management. During this phase an asset owner will assign a value of criticality which ranges from 1 to 10, with 10 being maximum. This value of criticality is assigned on the

basis of importance of confidentiality, integrity and availability of that device. To perform our analysis we have assumed the following sample values of 15 gateways:-

Table 3.1: Sample Values of Criticality, Size and Traffic for 15 Gateways

Gateway	Criticality	Size	Traffic (KB)
GW-1	5	152	800
GW-2	5	175	900
GW-3	7	290	800
GW-4	3	79	400
GW-5	9	150	295
GW-6	10	85	340
GW-7	1	93	1200
GW-8	2	298	300
GW-9	6	300	340
GW-10	4	50	743
GW-11	7	300	900
GW-12	2	175	912
GW-13	3	70	1200
GW-14	8	120	498
GW-15	5	165	351

3.4.2.1 Criticality

In some organizations criticality could be a single and most important factor to secure any IoT asset. For example in a nuclear reactor an IoT gateway providing connection to the sensors at atomic reactors is more critical than the one which is providing connections to the lighting and Air conditioning systems in the dining hall of the same reactor. Such organizations would always update the firmware and install patches on the critical gateways irrespective of the traffic and number of connected devices with that gateway. If we arrange different gateways in descending order of the criticality then the list of our sample data will be as depicted in Table 3.2 and graphical representation in figure: 3.2:-

Table 3.2: List of 15 Gateways Arranged in Criticality Descending Order

Serial No.	Gateway	Criticality	Size	Traffic(KB)
1	GW- 6	**10**	85	340
2	GW- 5	**9**	150	295
3	GW- 14	**8**	120	498
4	GW- 11	7	300	900
5	GW- 3	7	290	800
6	GW- 9	6	300	340
7	GW- 2	5	175	900
8	GW- 1	5	152	800
9	GW- 15	5	165	351
10	GW- 10	4	50	743
11	GW- 13	3	70	1200
12	GW- 4	3	79	400
13	GW- 12	2	175	912
14	GW- 8	2	298	300
15	GW- 7	1	93	1200

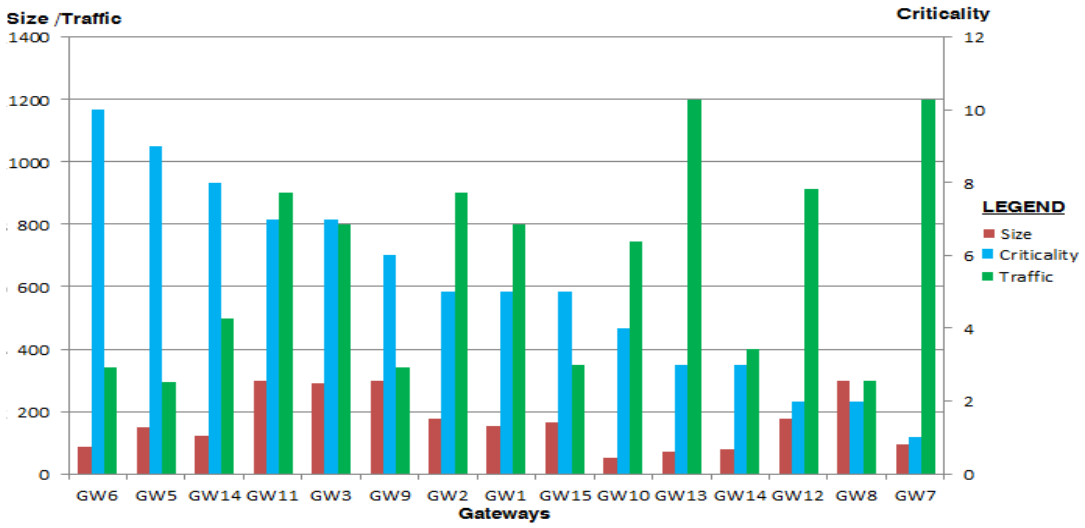


Figure. 3.2: Graph of Values Arranged in Criticality Descending Order

In Table: 3.2, **gateways-6,5 and 14** will be patched first as they have higher criticality value. We can see that **gateways-13,7, and 12** have higher network traffic and low criticality value so they will not be patched. Similarly **gateways 11,9 and 8** have larger number of connected devices but low criticality value. So these gateways will not be patched. The **graph shows maximum values in criticality of a gateway while minimum values of size and traffic** of the same gateway. So arranging gateways only on the basis of criticality will not be a good idea. We assume that **criticality** is most

important factor to determine the security of any Information asset including IoT. The **value of criticality** of IoT gateways will be determined beforehand. Criticality of IoT gateway will depend on three security services—confidentiality, integrity and availability.

3.4.2.2 Size of the Network

Size of a network means the number of connected devices to the gateway. We assume that IoT devices remain on for 24/7, therefore this figure will be close to the actual number of devices in that subnet. Size of a network is an important parameter. Large networks will have huge traffic volumes. Such networks will be more prone to attacks. If size of the network is large and it is already infected it is more dangerous to launch attack against other targets. If we arrange our gateways with respect to size we get Table 3.3 and graph in Fig. 3.3.

Table 3.3: List of 15 Gateways Arranged in Size Descending Order

Serial No.	Gateway	Criticality	Size	Traffic(KB)
1	GW-11	7	**300**	900
2	GW-9	6	**300**	340
3	GW-8	2	**298**	300
4	GW-3	7	290	800
5	GW-2	5	175	**900**
6	GW-12	2	175	912
7	GW-15	5	165	351
8	GW-1	5	152	800
9	GW-5	**9**	150	295
10	GW-14	**8**	120	498
11	GW-7	1	93	**1200**
12	GW-6	**10**	85	340
13	GW-4	3	79	400
14	GW-13	3	70	**1200**
15	GW-10	4	50	743

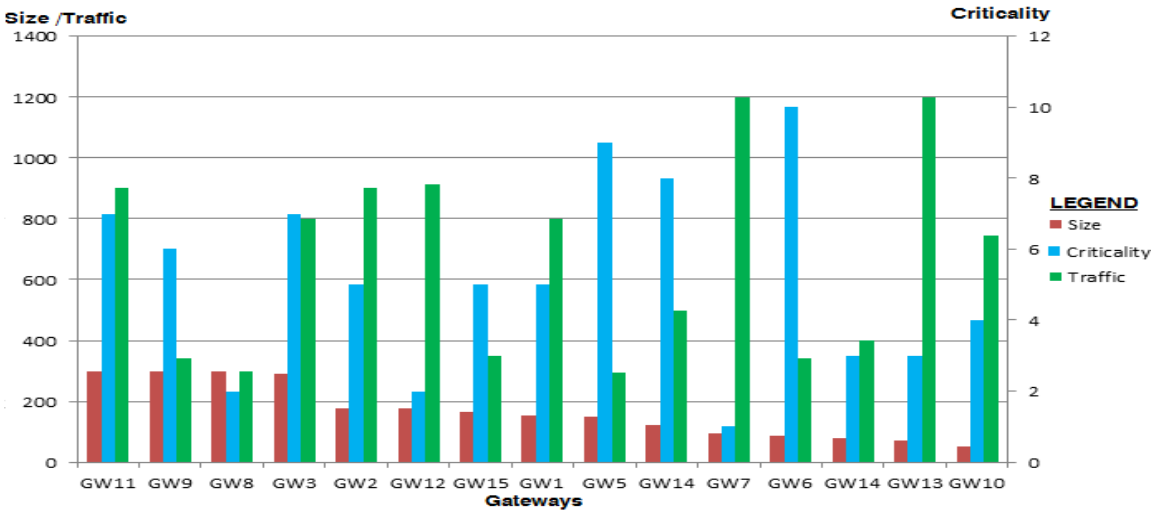


Figure. 3.3: Graph of Values Arranged in Size Descending Order

If we have only three number of patches then in above Table 3.3 **gateways-11,9 and 8** will be patched, since they have maximum number of connected devices. If the organization also gives importance to criticality and network traffic then above patching criteria will not work since **gateway-6** with maximum criticality value i.e.10 is at **12th number** in this dataset of 15 gateways. Similarly **gateways-7 and 13** with maximum network traffic are at **11th and 14th** position in the patching order of gateways.

3.4.2.3 Network Traffic

Network Traffic is very important parameter which is equally considered both in network performance and security. S.M Cheng *et al* [6] proposed a patching scheme in which they considered only the volume of network traffic as a criteria to patch the gateways. According to their scheme gateways with higher volume of network traffic will be patched first. We consider that **network traffic is dependent on the number of active devices connected with a gateway**. We also suggest that network traffic is not as important as the other two a/m parameters due to following reasons:-

- A network with high traffic can be **less important** than another critical network. Simple example is that in a particular situation one would give more importance to the door locks of a building than the lighting system of the same building because any compromise of the door locks would provide an easy access to intruders.
- A high volume of network traffic could be a **divisionary attack**.

If an organization is only interested to prioritize its security settings according to network traffic then our list of gateways arranged in descending order of network traffic is shown in Table 3.4 and relevant graph in figure: 3.4 as under:-

Table 3.4: List of 15 Gateways Arranged in Traffic Descending Order

Serial No.	Gateway	Criticality	Size	Traffic
1	GW-7	1	93	**1200**
2	GW-13	3	70	**1200**
3	GW-12	2	175	**912**
4	GW-11	7	**300**	900
5	GW-2	5	175	**900**
6	GW-3	7	290	800
7	GW-1	5	152	800
8	GW-10	4	50	743
9	GW-14	**8**	120	498
10	GW-4	3	79	400
11	GW-15	5	165	351
12	GW-9	6	**300**	340
13	GW-6	**10**	85	340
14	GW-8	2	**298**	300
15	GW-5	**9**	150	295

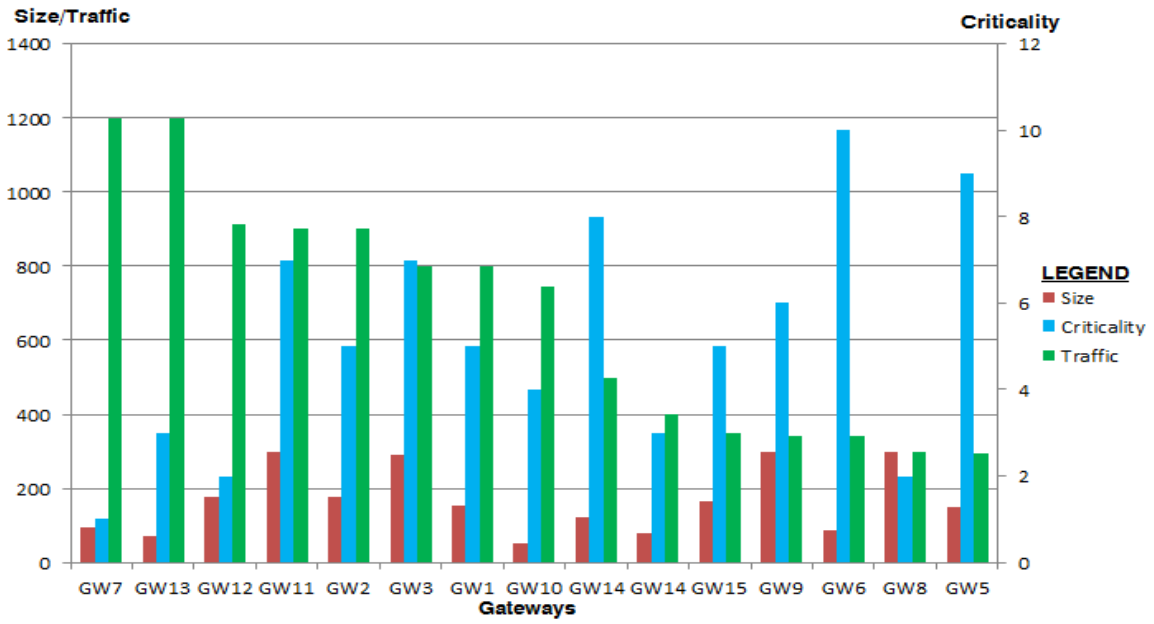


Figure. 3.4: Graph of Values Arranged in Traffic Descending Order

In Table 3.4 above we can see that only **gateways-7,13 and 12** with highest network traffic will be patched first. These gateways have **criticality values 1,3 and 2** respectively which is lowest in the list. Similarly our **gateways-11,9 and 8** with higher value of network size and **gateways-6,5 and 14** with higher values of criticality will not be patched. So if we consider only network traffic as criteria for patching we can miss other important parameters of an IoT network.

Above results show that only a **single factor from traffic, criticality or size of a network will not lead to a successful patching scheme or ordered list of gateways**. Therefore we need to device a function which takes values of all three parameters as input and generate a single value. Here we are interested in gateways with higher values of such function. We call this as our objective function which has been explained under Para 3.4.4.

3.4.3 Network Load

We know that network traffic is dependent on the network size i.e. number of devices in a network. For example **gateway-A with 30 connected** IoT devices will generate high network traffic than **gateway-B with 10 connected devices** in the same environment. So according to S.M Cheng *et al* [6] **gateway-A(with 30 devices)** with higher traffic volume will be patched first than the **gateway-B(with 10 devices)** and less network traffic. But what if **gateway-B (with 10 devices)** is producing network traffic **850KB** in a specific time which is just less than **900KB** traffic being produced by **gateway-A(with 30 devices)**. In such case patching scheme should first patch gateway-B which is producing high traffic with less number of connected devices. Such behavior makes gateway-B suspicious and best candidate for patching. Therefore we device a new parameter "*Network Load*" which is dependent on size (number of connected devices) and network traffic.

$$Load(l) = traffic(t)/Size(s)$$

If we put the values in above equation then we get following result:-

Load(l) of gateway-A = 900/30= ***30***

Load(l) of gateway-B = 850/10= ***85***

Load of gateway-B is higher than the load of gateway-A. So gateway-B should be patched first. Therefore we will use network load along with criticality in in our patching scheme to decide which IoT gateways to be patched first.

3.4.4 Objective Function

Our objective function ***f(c,l)*** takes values of two important parameters of IoT network i.e. **criticality and load** (traffic/size) as input. Assigns different weightages to these parameters by multiplying the value with its weightage and adding the two. Objective function is given as under:-

$$f(c, l) = c_w(c) + l_w(l)$$

c and l are values for criticality and load of a gateway.

c_w and l_w are the weightages for criticality and load.

We have analyzed different weightages for criticality and load. Comparison is attached as Appendix-"A". Since criticality of a network is very important therefore we start with assigning **80%** weightage to criticality. Therefore the values for two constants (c_w and l_w) for objective function are given as under:-

Weightage of criticality (c_w) = **0.8 (80%)** [range of criticality value= 1-10]

Weightage of Load (l_w) = **0.2 (20%)**

Now we calculate the value of objective function on our dataset and then arrange the list of gateways in descending order of the objective function.

Table 3.5: List of Gateways in descending order of the Value of Objective Function

(80% weightage for criticality and 20% weightage for load)

Serial No.	Gateway	Criticality	Size	Traffic	Load	Objective Function
1	GW-6	**10**	85	340	4	**8.8**
2	GW-5	**9**	150	295	1.966666667	**7.593333333**
3	GW-14	**8**	120	498	4.15	**7.23**
4	GW-11	7	300	900	3	6.2
5	GW-10	4	50	743	14.86	6.172
6	GW-3	7	290	800	2.75862069	6.151724138
7	GW-13	3	70	1200	17.14285714	5.828571429

8	GW-1	5	152	800	5.263157895	5.052631579
9	GW-2	5	175	900	5.142857143	5.028571429
10	GW-9	6	300	340	1.133333333	5.026666667
11	GW-15	5	165	351	2.127272727	4.425454545
12	GW-4	3	79	400	5.063291139	3.412658228
13	GW-7	1	93	1200	12.90322581	3.380645161
14	GW-12	2	175	912	5.211428571	2.642285714
15	GW-8	2	298	300	1.006711409	1.801342282

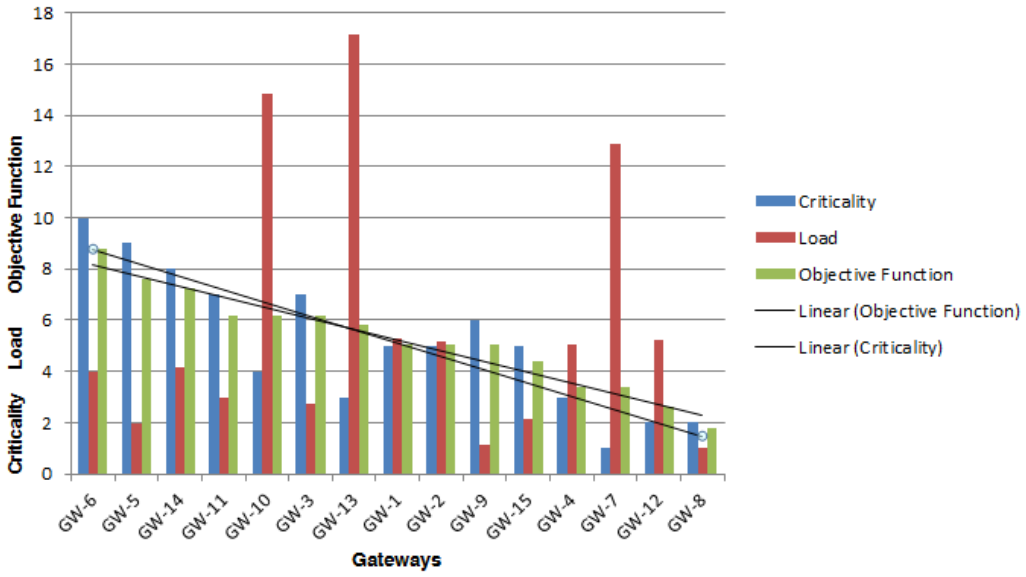


Figure. 3.5: Graph of Values Arranged in Objective Function Descending Order
(80% weightage for criticality and 20% weightage for load)

The results show that if we assign **80% weightage** to criticality then ordered list of gateways is almost **similar to the list in Table 3.2** which is the list of gateways arranged w.r.t the value of criticality. If we plot the values of criticality, load and objective function we can see the pattern (trend lines in figure: 3.5) of criticality and objective function is almost similar. So here we deduce that **if we assign 80% value to criticality then we do not get our desired list which gives adequate importance to other parameters as well**.

Now, we assign **60%** weightage to **criticality**. Therefore the values for two constants (*cw* and (*lw*) for objective function are given as under:-

Weightage of criticality (*cw*) = **0.6 (60%)** [range of criticality value= 0-10]

Weightage of load (*lw*) = **0.4 (40%)**

Table 3.6 shows the values of objective function in descending order with new weightage assigned to criticality and load:-

Table 3.6: List of Gateways in Descending order of the Value of Objective function

(60% weightage for criticality and 40% weightage for load)

Gateway	Criticality	Size	Traffic	Load	Objective Function
GW-13	3	70	1200	17.14285714	8.657142857
GW--10	4	50	743	14.86	8.344
GW-6	**10**	85	340	4	7.6
GW-14	**8**	120	498	4.15	6.46
GW-5	**9**	150	295	1.966666667	6.186666667
GW-7	1	93	1200	12.90322581	5.761290323
GW-11	7	300	900	3	5.4
GW-3	7	290	800	2.75862069	5.303448276
GW-1	5	152	800	5.263157895	5.105263158
GW-2	5	175	900	5.142857143	5.057142857
GW-9	6	300	340	1.133333333	4.053333333
GW-15	5	165	351	2.127272727	3.850909091
GW-4	3	79	400	5.063291139	3.825316456
GW-12	2	175	912	5.211428571	3.284571429
GW-8	2	298	300	1.006711409	1.602684564

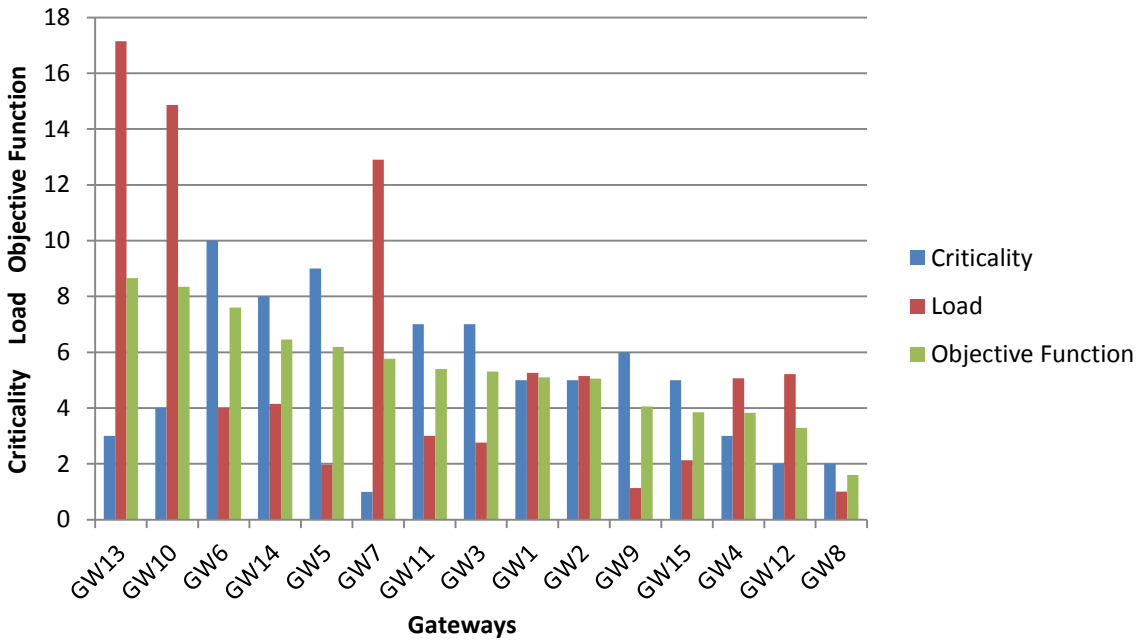


Figure. 3.6: Graph of Values Arranged in Objective Function Descending Order

(60% weightage for criticality and 40% weightage for load)

Table 3.6 shows that if gateways are arranged in descending order of the value of objective function then we get **desired result**. One of the top three gateways (**gateway-6**) has maximum criticality value of **10**. Other gateway (**gateway-13**) has maximum network traffic (**1200KB**) and maximum load (**17.14**) at the same time. **Gateway-10** from the top three gateways has **second highest value of network load(14.86)**. Top three **gateways 13,10 and 6 with higher value of objective function** will be patched first. Above results show that important parameters of IoT network i.e. criticality, traffic and size have been catered for. This scheme can be applied in any network where administrators can arrange the gateways in an order. This ordered list of gateways can be used as a patching sequence.

We conclude that formula for our objective function and weightages for criticality and load constants is given as under:-

$$f(c, l) = c_w(c) + l_w(l)$$

c and l are values for criticality and load of a gateway.

c_w and l_w are the weightages for criticality and load.

Values $c_w = 0.6$, $l_w = 0.4$ are constants.

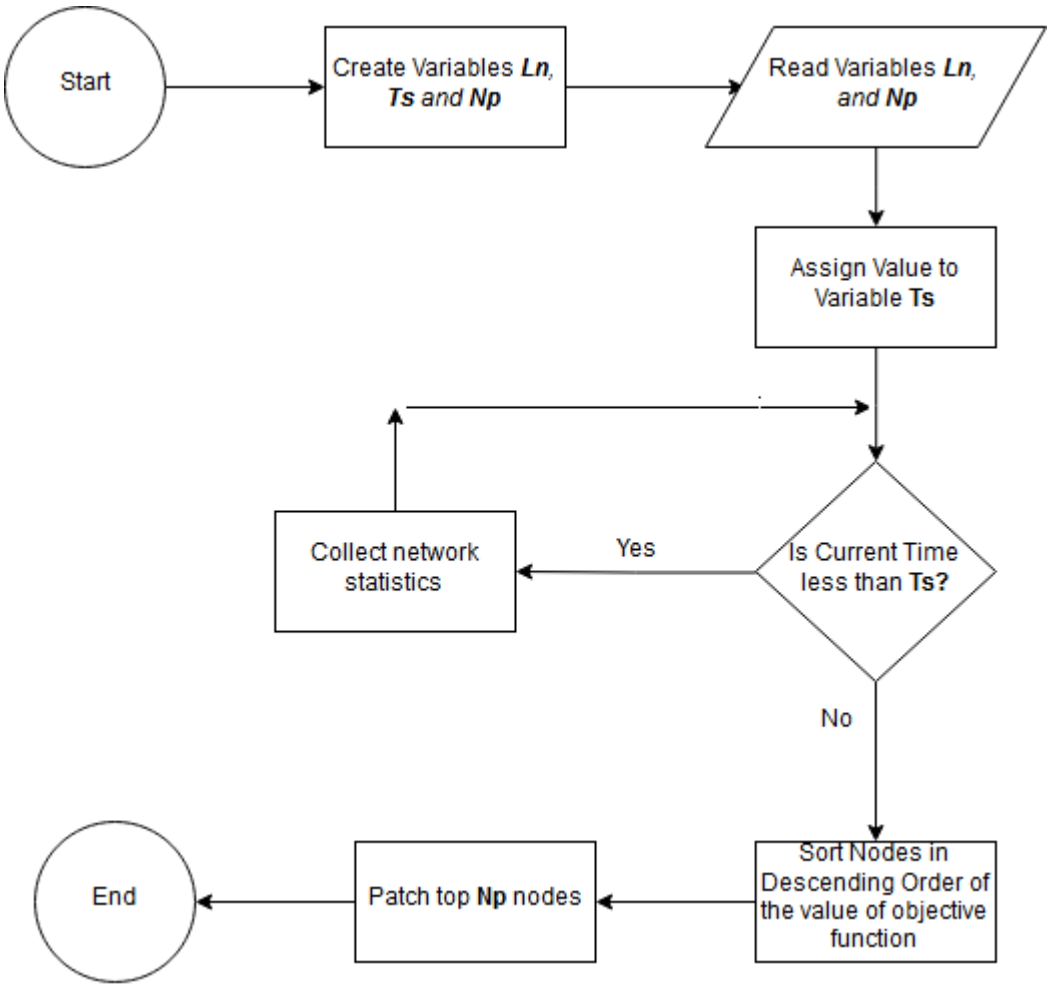


Figure. 3.7: Flow Chart of Patching Algorithm

(Ts: Time to start patching, Ln: List of nodes to be patched, Np: Number of patches)

Patching Algorithm

Inputs: *List of nodes to be patched (**Ln**), time to start patching (**Ts**), Number of available patches patches(**Np**)*

1. ***Start***
2. *Declare variables **Ln**, **Ts** and **Np***
3. *Read variables **Ln** and **Np***
4. *Assign value current time **+ 2** to variable **Ts***
5. ***if*** *current time < **Ts*** ***then***
6. *collect network statistics*
7. ***else***
8. *Sort the nodes in descending order of the value of objective function*
9. *Patch top **Np** nodes*
10. ***end if***
11. ***Stop***

Figure. 3.8: Patching Algorithm

Chapter 4

CONCLUSION

4.1 Introduction

A pervasive technology like IoT which has a potential of exponential growth is always marred with certain issues. Industries which want to stay relevant, progress and eye on future growth will have to embrace the technology with all its pros and cons. Then there is a continuous tug of war between the bad and good guys. There are technologies to establish a scalable, efficient and secure network. But security is a collective or community responsibility. All relevant stakeholders i.e. users, manufacturers and government or regulators have to play their role and beat the bad guys collectively.

In an IoT network a smart gateway will continue to play an important role. It not only works as a bridge between the constrained devices and Internet cloud but also analyses data received from sensors before sending it to cloud. If gateway is regularly patched, it can mitigate or stop the propagation of malware. This technique is extremely useful in IoT where it is difficult to manually patch huge number of heterogeneous devices with different types of patches. A large organization with its network spread over different cities with hundreds of gateways or BS can face a problem i.e. which gateways to be patched first specially when there are limited number of patches available. In this thesis we have provided a gateway patching scheme with a function that takes account of all important parameters of a network i.e. criticality, traffic and size/load and gives a value as output. Gateways are organized in descending order of the output value. Top gateways with highest output value are patched first. We have analyzed that the objective function in our patching scheme gives due importance to all important parameters.

4.1 Future Work

Patching of gateway can only stop the spread of malware from one subnet to another. Malware will still remain within the network. The next part of this scheme could be how to patch the devices?

In this scheme we have used 2 minutes time in which gateway traffic and other statistics are collected. This time can be reduced, which will mean that we are quickly stopping the spread hence less harm. On the other hand if we reduce monitoring time, it is possible that we may not get statistics which are close to reality.

BIBLOGRAPHY

[1] Gartner Press Release, "Gartner Says 8.4 Billion Connected 'Things' Will Be in Use in 2017, Up 31 Percent From 2016. Internet: https://www.gartner.com/newsroom/id/3598917, Feb. 7, 2017 [Dec. 28, 2017]

[2] Constantinos Kolias, Georgios Kambourakis, Angelos Stavrou, Jeffrey Voas., "DDoS in the IoT: Mirai and other Botnets" IEEE *Computer*, Volume: 50, Issue: 7, pp. 80-84, 2017.

[3] Nicky Woolf, "DDoS attack that disrupted internet was largest of its kind in history, experts say", Internet: https://www.theguardian.com/technology/2016/oct/26/ddos-attack-dyn-mirai-botnet , Oct. 26, 201 [May. 11, 2018].

[4] Arunan Sivanathan, Daniel Sherratt, Hassan Habibi Gharakheili, AdamRadford, Chamith Wijenayake, Arun Vishwanath, and Vijay Sivaraman,"Characterizing and Classifying IoT Traffic in Smart Cities and Campuses". *2017 IEEE Conference on Computer Communications Workshops*.

[5] B. Kang and H. Choo, "An experimental study of a reliable IoT gateway" , ICT Express, 2017.

[6] S. M. Cheng, P. Y. Chen, C. C. Lin, and H. C. Hsiao, "Traffic-Aware Patching for Cyber Security in Mobile IoT," IEEE Communications Magazine, vol. 55, no. 7, pp. 29–35, 2017.

[7] Knud Lasse Lueth, "Why the Internet of Things is called Internet of Things: Definition, history, disambiguation." Internet: https://iot-analytics.com/internet-of-things-definition/, Dec. 19, 2014 [Jul. 29, 2018].

[8] Karen Rose, Scott Eldridge, Lyman Chapin "The Internet of Things (IoT): An Overview Understanding the Issues and Challenges of a More Connected World" https://www.internetsociety.org/wp-content/uploads/2017/08/ISOC-IoT-Overview-20151221-en.pdf , Oct. 15, 2015 [Jun. 9, 2018].

[9] Greg Dunko, Joydeep Misra, Josh Robertson and Tom Snyder (Mar. 1, 2017), *A Reference Guide to the Internet of Things* (1st edition) [On-line]. Available: https://bridgera.com/ebook/ [June 13, 2018].

[10] Roy Want, Bill N. Schilit, Scot Jenson. (Jan. 2015.). "Enabling the Internet of Things." *Computer*. [On-line]. 48(1), pp. 28-35. Available: https://ieeexplore.ieee.org/abstract/document/7030240/ [Jun. 9, 2018].

[11] Richard Barry, "Direct-to-device connectivity in the Internet of Things." Internet: https://www.embedded.com/design/real-world-applications/4426949/Direct-to-device-connectivity-in-the-Internet-of-Things, Jan. 12, 2014 [Jun. 8, 2018]

[12] Cahit Akin, "IoT: Centralized Vs. Distributed Architectures," Information Week Network Computing, Available: http://www.networkcomputing.com/networking/iot-centralizedvs-distributed-architectures/435583941, 2015.

[13] Y. Saied, A. Olivereau, D. Zeghlache and M. Laurent, “Trust management system design for the internet of things: a context-aware and multi-service approach,” Elsevier, Computer Security 39 (2013) 351– 365, 2013.

[14] Tom Bradicich,“Exploring the Four Stages of an Industrial IoT Solution”. Internet: https://community.hpe.com/t5/Internet-of-Things-IoT/Exploring-the-Four-Stages-of-an-Industrial-IoT-Solution/ba-p/6917607#.WxjMjIpRXIU, Nov. 16, 2016 [Jun. 7, 2018].

[15] Jayavardhana Gubbi, Rajkumar Buyya, Slaven Marusic and Marimuthu Palaniswami, “Internet of Things (IoT): A vision, architectural elements, and future directions”, *Future Generation Computer Systems*, vol. 29, pp. 1645-1660, 2013.

[16] JR Fuller, “The 4 stages of an IoT architecture”. Internet: https://techbeacon.com/4-stages-iot-architecture, May. 26, 2016 [Jun. 7, 2018].

[17] Rafael Rocha, “The IoT Architecture at the Edge”. Internet: https://www.iotcentral.io/blog/the-iot-architecture-at-the-edge, March. 3, 2017 [Jun. 14, 2018].

[18] i-scoop, “What is the Internet of Things? Internet of Things definitions” Internet: https://www.i-scoop.eu/internet-of-things/ [Jul. 30, 2018]

[19] Zhi-Kai Zhang, Michael Cheng Yi Cho, Chia-Wei Wang, Chia-Wei Hsu, Chong-Kuan Chen, Shiuhpyng Shieh, “IoT Security: Ongoing Challenges and Research Opportunities”, 2014 IEEE 7th International Conference on Service-Oriented Computing and Applications, 2014, 2163-2871.

[20] Jun Zhou, Zhenfu Cao, Xiaolei Dong,Athanasios V. Vasilakos, “Security and Privacy for Cloud-Based IoT: Challenges, Countermeasures, and Future Directions”. *IEEE Communications Magazine*, vol 55(1), pp. 26-33, Jan. 2017.

[21] John Leyden. “Biggest DDoS Attack in history hammers Spamhaus” Internet: https://www.theregister.co.uk/2013/03/27/spamhaus_ddos_megaflood/, Mar. 27, 2013 [Jul 28, 2018].

[22] Parmy Olson “The Largest Cyber Attack In History Has Been Hitting Hong Kong Sites.” Internet: https://www.forbes.com/sites/parmyolson/2014/11/20/the-largest-cyber-attack-in-history-has-been-hitting-hong-kong-sites/, Nov. 20, 2014 [Jul. 28, 2018].

[23] Maria Korolov “DDoS attack on BBC may have been biggest in history.” Internet: https://www.csoonline.com/article/3020292/cyber-attacks-espionage/ddos-attack-on-bbc-may-have-been-biggest-in-history.html, Jan. 8, 2016 [Jul. 29, 2018].

[24] Michael Bailey, Evan Cooke, Farnam Jahanian, Yunjing Xu, Manish Karir, “A Survey of Botnet Technology and Defenses”. IEEE CATCH, Washington, DC, USA, 2009.

[25] *Radware*, “A Quick History of IoT Botnets” https://blog.radware.com/uncategorized/2018/03/history-of-iot-botnets/ Mar. 1, 2018, [Aug 10, 2018].

[26] Matt Toomey, “IoT Device Security is Being Seriously Neglected” . Internet: https://www.business2community.com/cybersecurity/iot-device-security-seriously-neglected-02007534 , Feb. 13, 2018 [Aug 10, 2018].

[27] WIND River Systems, “Internet of Things Security Is More Challenging Than Cybersecurity” (White Paper), 2016.

[28] NIST, “Framework for Improving Critical Infrastructure Cybersecurity”, Internet: https://nvlpubs.nist.gov/nistpubs/CSWP/NIST.CSWP.04162018.pdf Apr. 16, 2018 [Aug 10, 2018].

COMPARISON OF VALUES

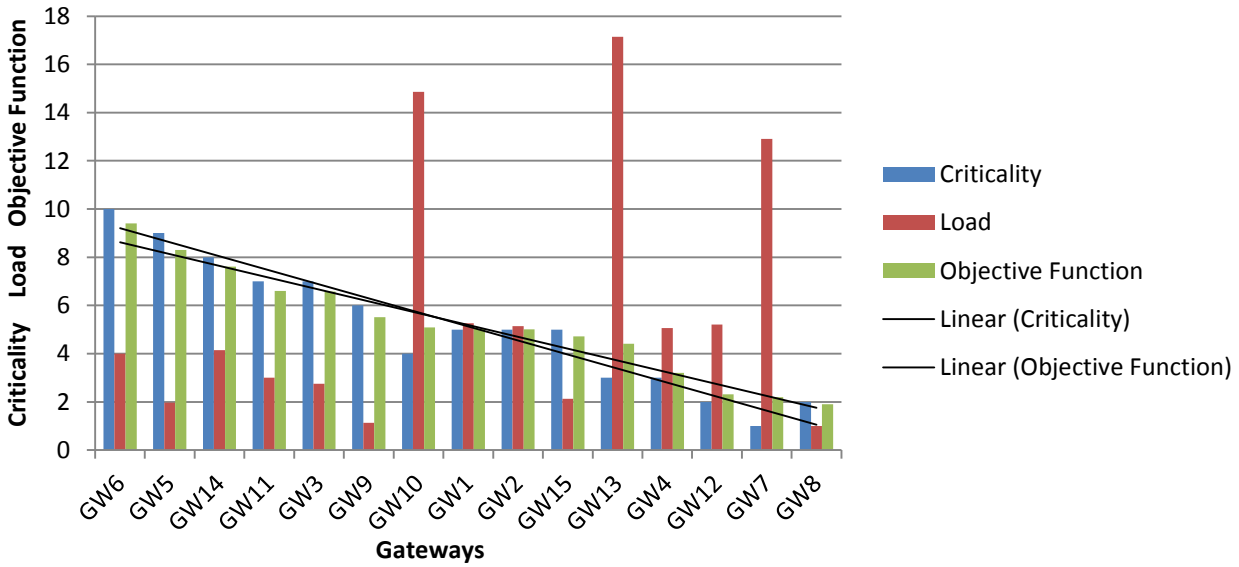
List of Gateways in descending order of the Value of Objective Function

(**90% weightage for criticality and 10% weightage for load in objective function**)

Objective Function $f(c, l) = c_w(c) + l_w(l)$

c and l are values of criticality and load. c_w=0.9 and l_w=0.1 are weightages of criticality and load in objective function

Gateway	Criticality	Size	Traffic	Load	Objective Function $f(c, l)$
GW6	**10**	85	340	4	9.4
GW5	**9**	150	295	1.966666667	8.296666667
GW14	**8**	120	498	4.15	7.615
GW11	7	300	900	3	6.6
GW3	7	290	800	2.75862069	6.575862069
GW9	6	300	340	1.133333333	5.513333333
GW10	4	50	743	14.86	5.086
GW1	5	152	800	5.263157895	5.026315789
GW2	5	175	900	5.142857143	5.014285714
GW15	5	165	351	2.127272727	4.712727273
GW13	3	70	1200	17.14285714	4.414285714
GW4	3	79	400	5.063291139	3.206329114
GW12	2	175	912	5.211428571	2.321142857
GW7	1	93	1200	12.90322581	2.190322581
GW8	2	298	300	1.006711409	1.900671141



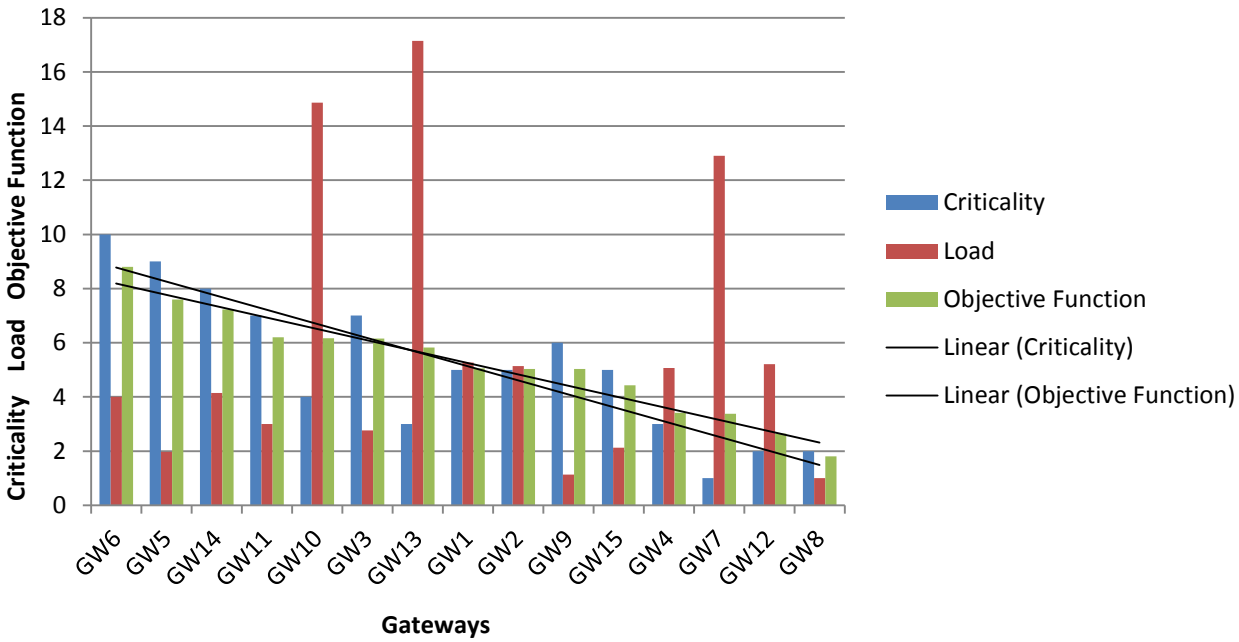
Above graph shows almost similar trend lines for Criticality and Objective Function.

List of Gateways in descending order of the Value of Objective Function
(80% weightage for criticality and 20% weightage for load in objective function)

Objective Function $f(c, l) = c_w(c) + l_w(l)$

c and l are values of criticality and load. c_w=0.8 and l_w=0.2 are weightages of criticality and load in objective function

Gateway	Criticality	Size	Traffic	Load	Objective Function $f(c, l)$
GW6	**10**	85	340	4	8.8
GW5	**9**	150	295	1.966666667	7.593333333
GW14	**8**	120	498	4.15	7.23
GW11	7	300	900	3	6.2
GW10	4	50	743	14.86	6.172
GW3	7	290	800	2.75862069	6.151724138
GW13	3	70	1200	17.14285714	5.828571429
GW1	5	152	800	5.263157895	5.052631579
GW2	5	175	900	5.142857143	5.028571429
GW9	6	300	340	1.133333333	5.026666667
GW15	5	165	351	2.127272727	4.425454545
GW4	3	79	400	5.063291139	3.412658228
GW7	1	93	1200	12.90322581	3.380645161
GW12	2	175	912	5.211428571	2.642285714
GW8	2	298	300	1.006711409	1.801342282



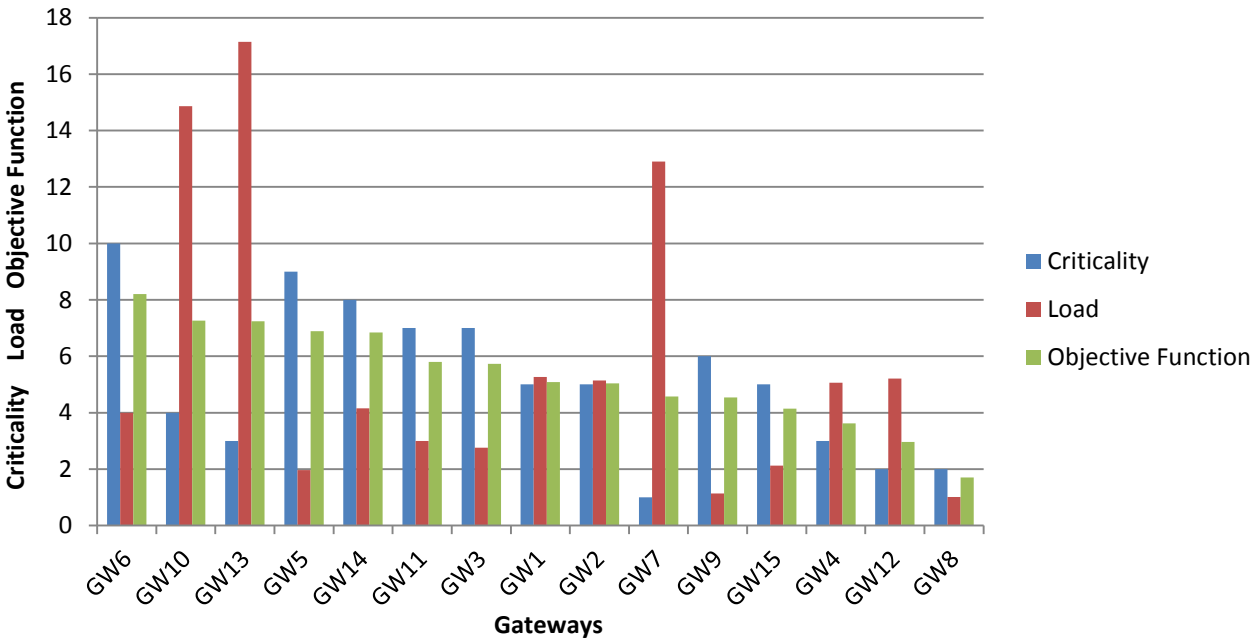
We have similar patterns of Criticality and Objective Function bars in the graph.

List of Gateways in descending order of the Value of Objective Function
(70% weightage for criticality and 30% weightage for load in objective function)

Objective Function $f(c, l) = c_w(c) + l_w(l)$

c and l are values of criticality and load. c_w=0.7 and l_w=0.3 are weightages of criticality and load in objective function

Gateway	Criticality	Size	Traffic	Load	Objective Function $f(c, l)$
GW6	**10**	85	340	4	8.2
GW10	4	50	743	14.86	7.258
GW13	3	70	1200	17.14285714	7.242857143
GW5	**9**	150	295	1.966666667	6.89
GW14	**8**	120	498	4.15	6.845
GW11	7	300	900	3	5.8
GW3	7	290	800	2.75862069	5.727586207
GW1	5	152	800	5.263157895	5.078947368
GW2	5	175	900	5.142857143	5.042857143
GW7	1	93	1200	12.90322581	4.570967742
GW9	6	300	340	1.133333333	4.54
GW15	5	165	351	2.127272727	4.138181818
GW4	3	79	400	5.063291139	3.618987342
GW12	2	175	912	5.211428571	2.963428571
GW8	2	298	300	1.006711409	1.702013423



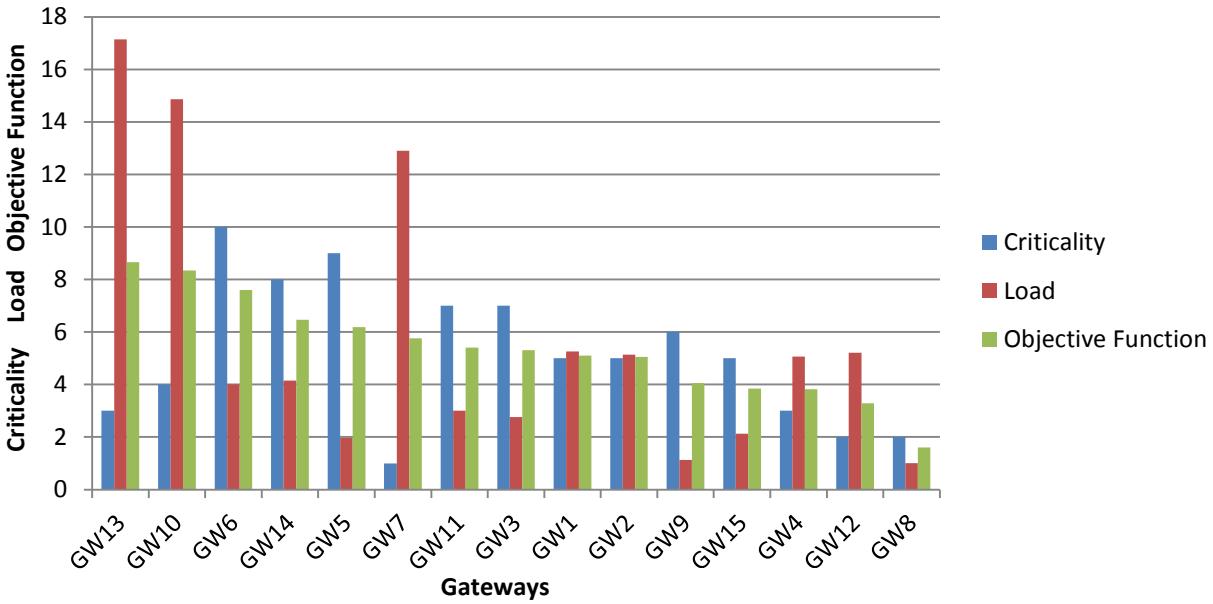
Above table shows that we have the list where one of top three gateways has highest criticality value of 10, One has highest network traffic 1200 KB and two gateways have the highest values of network load.

List of Gateways in descending order of the Value of Objective Function
(60% weightage for criticality and 40% weightage for load in objective function)

Objective Function $f(c, l) = c_w(c) + l_w(l)$

c and l are values of criticality and load. c_w=0.6 and l_w=0.4 are weightages of criticality and load in objective function

Gateway	Criticality	Size	Traffic	Load	Objective Function $f(c, l)$
GW13	3	70	1200	17.14285714	8.657142857
GW10	4	50	743	14.86	8.344
GW6	**10**	85	340	4	7.6
GW14	**8**	120	498	4.15	6.46
GW5	**9**	150	295	1.966666667	6.186666667
GW7	1	93	1200	12.90322581	5.761290323
GW11	7	300	900	3	5.4
GW3	7	290	800	2.75862069	5.303448276
GW1	5	152	800	5.263157895	5.105263158
GW2	5	175	900	5.142857143	5.057142857
GW9	6	300	340	1.133333333	4.053333333
GW15	5	165	351	2.127272727	3.850909091
GW4	3	79	400	5.063291139	3.825316456
GW12	2	175	912	5.211428571	3.284571429
GW8	2	298	300	1.006711409	1.602684564



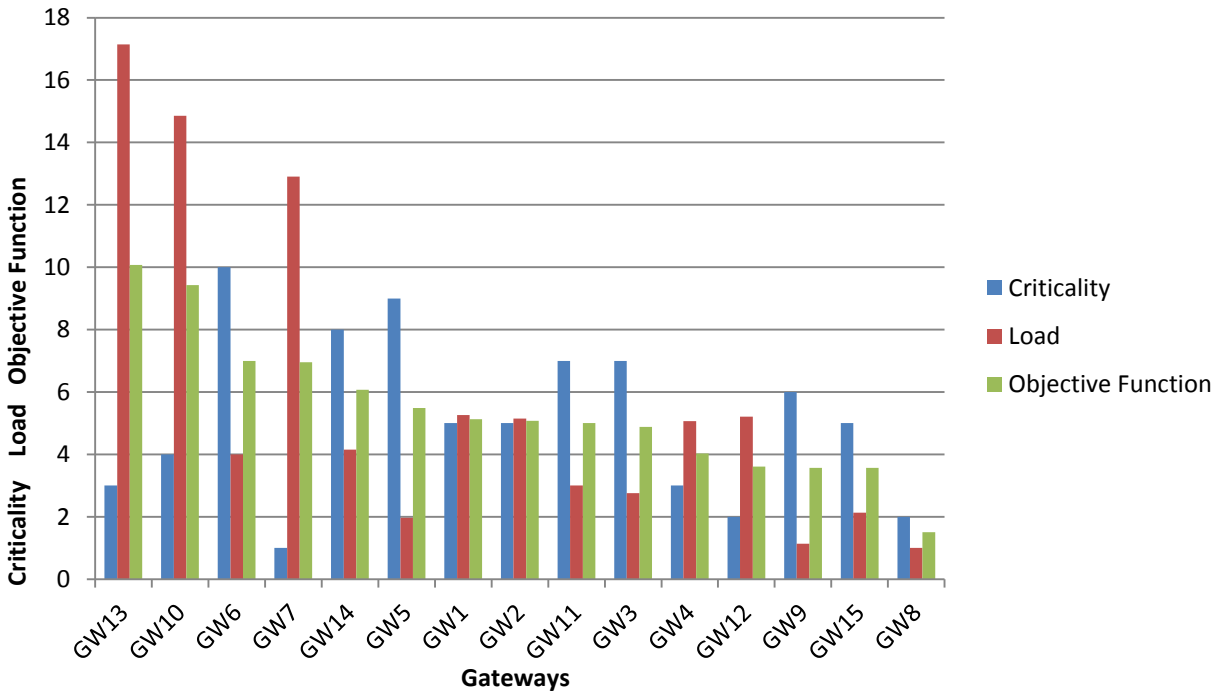
Above table and graph shows if we assign 60% weightage to Criticality and 40% weightage to network load then we get the desired list where top gateways have highest values of all important parameters like criticality, traffic and load.

List of Gateways in descending order of the Value of Objective Function
(50% weightage for criticality and 50% weightage for load in objective function)

Objective Function $f(c, l) = c_w(c) + l_w(l)$

c and l are values of criticality and load. c_w=0.5 and l_w=0.5 are weightages of criticality and load in objective function

Gateway	Criticality	Size	Traffic	Load	Objective Function $f(c, l)$
GW13	3	70	1200	17.14285714	10.07142857
GW10	4	50	743	14.86	9.43
GW6	**10**	85	340	4	7
GW7	1	93	1200	12.90322581	6.951612903
GW14	**8**	120	498	4.15	6.075
GW5	**9**	150	295	1.966666667	5.483333333
GW1	5	152	800	5.263157895	5.131578947
GW2	5	175	900	5.142857143	5.071428571
GW11	7	300	900	3	5
GW3	7	290	800	2.75862069	4.879310345
GW4	3	79	400	5.063291139	4.03164557
GW12	2	175	912	5.211428571	3.605714286
GW9	6	300	340	1.133333333	3.566666667
GW15	5	165	351	2.127272727	3.563636364
GW8	2	298	300	1.006711409	1.503355705



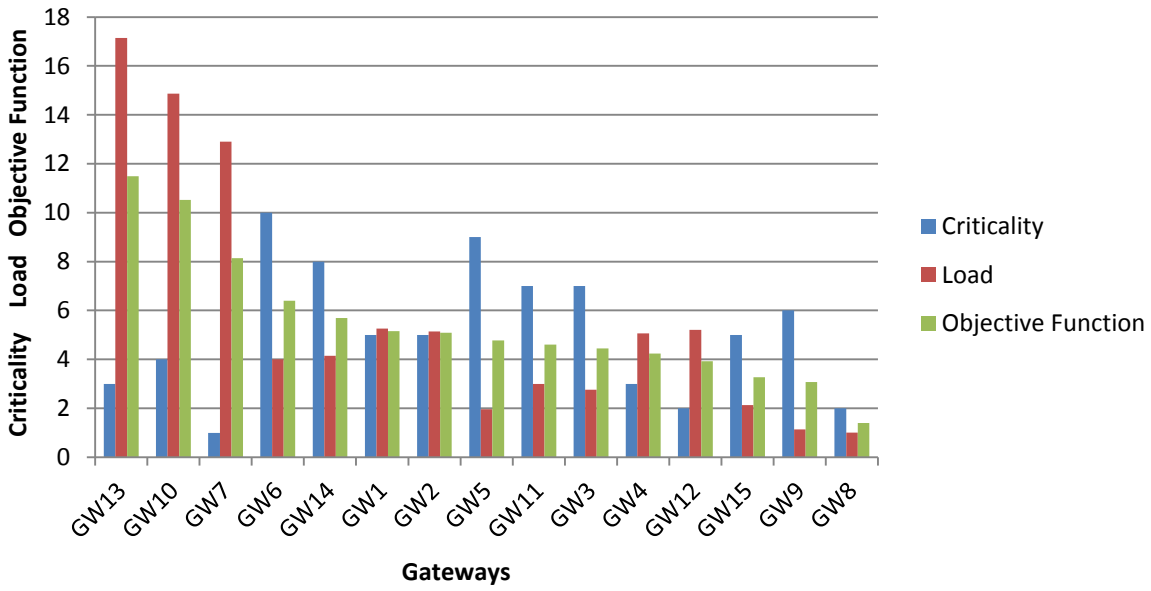
With increasing weightage of network load in objective function we see in the graph the pattern of Objective Function and network load is almost similar.

List of Gateways in descending order of the Value of Objective Function
(40% weightage for criticality and 60% weightage for load in objective function)

Objective Function $f(c, l) = c_w(c) + l_w(l)$

c and l are values of criticality and load. c_w=0.4 and l_w=0.6 are weightages of criticality and load in objective function

Gateway	Criticality	Size	Traffic	Load	Objective Function $f(c, l)$
GW13	3	70	1200	17.14285714	11.48571429
GW10	4	50	743	14.86	10.516
GW7	1	93	1200	12.90322581	8.141935484
GW6	**10**	85	340	4	6.4
GW14	**8**	120	498	4.15	5.69
GW1	5	152	800	5.263157895	5.157894737
GW2	5	175	900	5.142857143	5.085714286
GW5	**9**	150	295	1.966666667	4.78
GW11	7	300	900	3	4.6
GW3	7	290	800	2.75862069	4.455172414
GW4	3	79	400	5.063291139	4.237974684
GW12	2	175	912	5.211428571	3.926857143
GW15	5	165	351	2.127272727	3.276363636
GW9	6	300	340	1.133333333	3.08
GW8	2	298	300	1.006711409	1.404026846



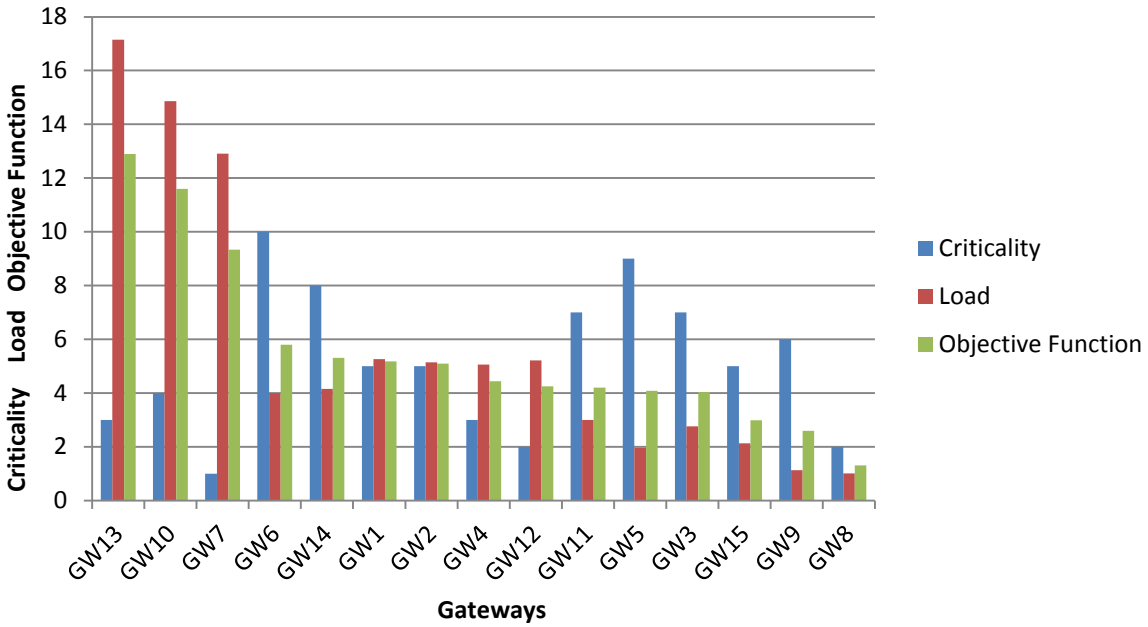
Above graph shows that top three gateways 13,10 and 7 have highest values of network load but no one among the top three has highest value of criticality.

List of Gateways in descending order of the Value of Objective Function
(30% weightage for criticality and 70% weightage for load in objective function)

Objective Function $f(c, l) = c_w(c) + l_w(l)$

c and l are values of criticality and load. c_w=0.3 and l_w=0.7 are weightages of criticality and load in objective function

Gateway	Criticality	Size	Traffic	Load	Objective Function $f(c, l)$
GW13	3	70	1200	17.14285714	12.9
GW10	4	50	743	14.86	11.602
GW7	1	93	1200	12.90322581	9.332258065
GW6	**10**	85	340	4	5.8
GW14	**8**	120	498	4.15	5.305
GW1	5	152	800	5.263157895	5.184210526
GW2	5	175	900	5.142857143	5.1
GW4	3	79	400	5.063291139	4.444303797
GW12	2	175	912	5.211428571	4.248
GW11	7	300	900	3	4.2
GW5	**9**	150	295	1.966666667	4.076666667
GW3	7	290	800	2.75862069	4.031034483
GW15	5	165	351	2.127272727	2.989090909
GW9	6	300	340	1.133333333	2.593333333
GW8	2	298	300	1.006711409	1.304697987



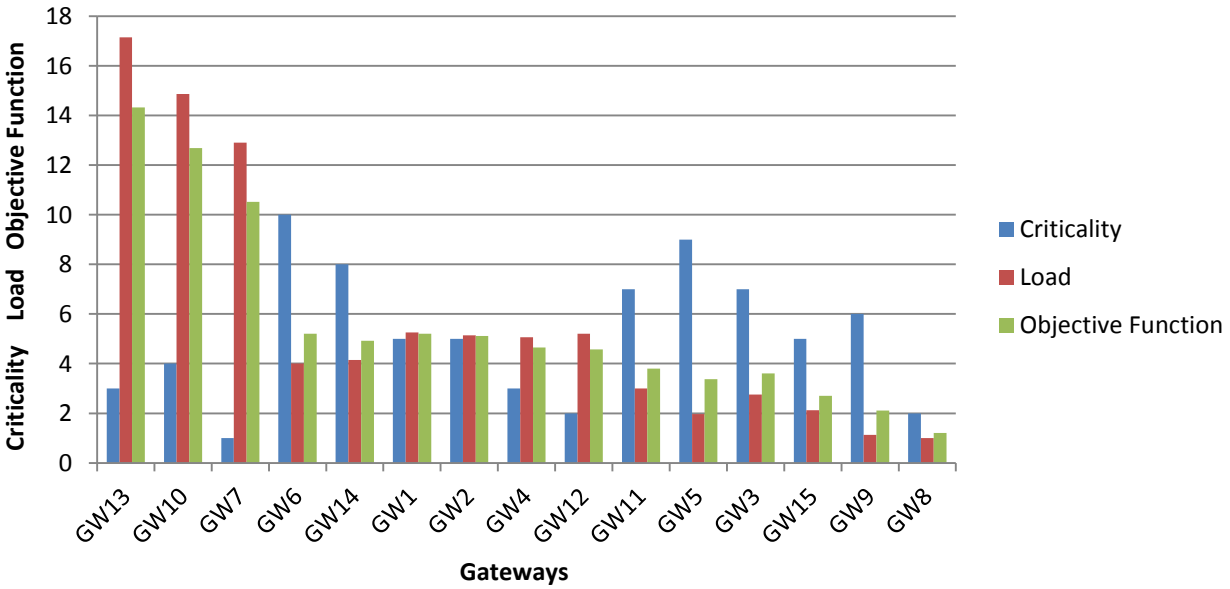
All top three gateways have highest values for load but no one among the top three has highest value of criticality.

List of Gateways in descending order of the Value of Objective Function
(30% weightage for criticality and 70% weightage for load in objective function)

Objective Function $f(c, l) = c_w(c) + l_w(l)$

c and l are values of criticality and load. c_w=0.2 and l_w=0.8 are weightages of criticality and load in objective function

Gateway	Criticality	Size	Traffic	Load	Objective Function $f(c, l)$
GW13	3	70	1200	17.14285714	14.31428571
GW10	4	50	743	14.86	12.688
GW7	1	93	1200	12.90322581	10.52258065
GW6	**10**	85	340	4	5.2
GW14	**8**	120	498	4.15	4.92
GW1	5	152	800	5.263157895	5.210526316
GW2	5	175	900	5.142857143	5.114285714
GW4	3	79	400	5.063291139	4.650632911
GW12	2	175	912	5.211428571	4.569142857
GW11	7	300	900	3	3.8
GW5	**9**	150	295	1.966666667	3.373333333
GW3	7	290	800	2.75862069	3.606896552
GW15	5	165	351	2.127272727	2.701818182
GW9	6	300	340	1.133333333	2.106666667
GW8	2	298	300	1.006711409	1.205369128



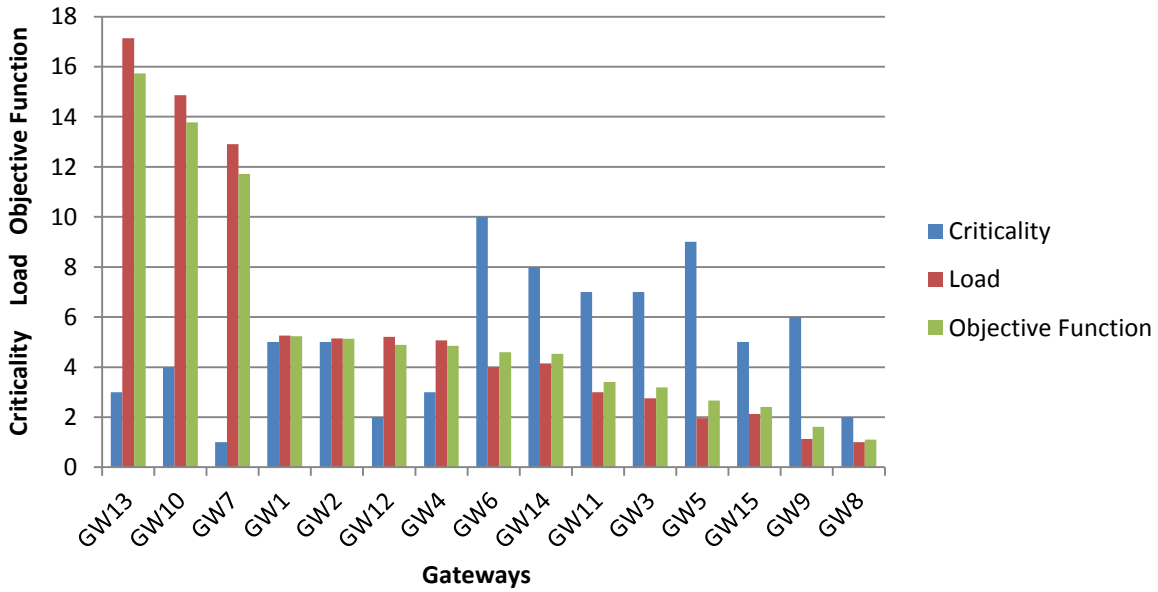
Above table and graph shows that with increasing weightage of load and decreasing the weightage of criticality we are having ordered list of gateways where gateways with highest network traffic and load are on the top. In the above table we see that no gateway among the top three is having the highest criticality value.

List of Gateways in descending order of the Value of Objective Function
(30% weightage for criticality and 70% weightage for load in objective function)

Objective Function $f(c, l) = c_w(c) + l_w(l)$

c and l are values of criticality and load. c_w=0.1 and l_w=0.9 are weightages of criticality and load in objective function

Gateway	Criticality	Size	Traffic	Load	Objective Function $f(c, l)$
GW13	3	70	1200	17.14285714	15.72857143
GW10	4	50	743	14.86	13.774
GW7	1	93	1200	12.90322581	11.71290323
GW1	5	152	800	5.263157895	5.236842105
GW2	5	175	900	5.142857143	5.128571429
GW12	2	175	912	5.211428571	4.890285714
GW4	3	79	400	5.063291139	4.856962025
GW6	**10**	85	340	4	4.6
GW14	**8**	120	498	4.15	4.535
GW11	7	300	900	3	3.4
GW3	7	290	800	2.75862069	3.182758621
GW5	**9**	150	295	1.966666667	2.67
GW15	5	165	351	2.127272727	2.414545455
GW9	6	300	340	1.133333333	1.62
GW8	2	298	300	1.006711409	1.106040268



Above table shows that we don't have the highest criticality value in top seven gateways. On the other hand all of the top three gateways have highest value of network load. So this criteria of weightages to different parameters is not realistic.