IJRET: International Journal of Research in Engineering and Technology eISSN: 2319-1163 | pISSN: 2321-7308

https://doi.org/10.15623/ijret.2018.0703010 Received: 19-01-2018, Accepted: 02-03-2018, Published: 22-03-2018

INTERNET OF THINGS: SECURITY ATTACKS AND SOLUTIONS

Ishank Pandey[1], Jagruti Karale[2], Sagar Thakare[3]

[1]NCRD's Sterling Institute of Management Studies, University of Mumbai
[2]NCRD's Sterling Institute of Management Studies, University of Mumbai
[3]NCRD's Sterling Institute of Management Studies, University of Mumbai

Abstract

The paper begins with the study of current status of IoT and inherent security concerns that it brings. IoT devices opens the scope for unlike devices to communicate and facilitate machine to machine(M2M) learning. Based on the already available literature, the author explains the vulnerability within the design, manufacturing and protocols of IoT and further attempts to correlate them with the cyber attack of the past.

----------***----------

1. INTRODUCTION

The Internet of things (IoT) is a collection of interconnected heterogeneous devices, services that can communicate over a wireless network to achieve a common objective in different areas.

The main objective of IoT is to make devices intelligent so that they could take care of their demands and supplies.

The IoT applications have seen rapid development in recent years due to the technologies of Radio Frequency identification (RFID) and Wireless Sensor Networks (WSN) [1]. The RFID enables the tagging or labeling of every single device, so as to serve as the basic identification mechanism in IoT. Due to WSN, each "thing" i.e. people, devices etc. becomes a wireless identifiable object and can communicate among the physical, cyber, and digital world.

Gartner, Inc. forecasts that 20.8 billion connected things will be in use worldwide in 2020 [2].

Table 1: Internet of Things Units Installed Base by Category (Millions of Units) [2].

Category	**2014**	**2015**	**2016**	**2020**
Consumer	2,277	3,023	4,024	13,509
Business: Cross-Industry	632	815	1,092	4,408
Business: Vertical-Specific	898	1,065	1,276	2,880
Grand Total	**3,807**	**4,902**	**6,392**	**20,797**

In terms of hardware spending, consumer applications will amount to $1534 billion in 2020, while the use of connected things in the enterprise will drive $1477 billion in 2020 [2].

Table 2: Internet of Things Endpoint Spending by Category (Billions of Dollars) [2].

Category	**2014**	**2015**	**2016**	**2020**
Consumer	257	416	546	1,534
Business: Cross-Industry	115	155	201	566
Business: Vertical-Specific	567	612	667	911
Grand Total	**939**	**1,183**	**1,414**	**3,010**

1.1 IoT Architecture

As the IoT is capable of connecting billions of heterogeneous objects via the Internet, there is an emerging requirement for a dynamic layered architecture.

Objects Layer: This is the first layer in IoT layered architecture. It is also known as perception layer [3]. This layer acquires data with the help of physical sensors of IoT to detect, collect and process the information.

Object Abstraction Layer: This layer is responsible for transferring data, which is acquired by Object layer, to the service management layer.

Data can be transferred via different technologies like 3G, 4G, GSM, UMTS, Wi-Fi, Bluetooth, ZigBee [3], etc.

Service Management Layer: This layer makes IoT application programmers to capable of working with heterogeneous objects.

Application Layer: This layer is responsible for high-quality smart services to retrieve what the customers want, according to their requirements. Eg : smart homes, smart production units, transportation, smart health care based biosensor equipment, etc.

Business Layer: This layer acquires data from application layer and according to that data it builds business model, graphs and flowcharts. This layer is responsible for managing the complete IoT system's activities and services.

https://doi.org/10.15623/ijret.2018.0703010 Received: 19-01-2018, Accepted: 02-03-2018, Published: 22-03-2018

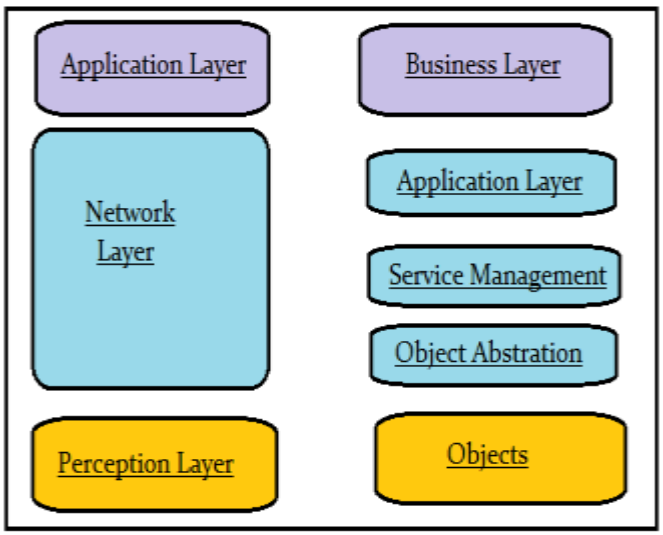


Fig 1.1: IoT Layered Architecture

1.2 Protocol

Some of the most important protocols are explained below which have defined by IEEE (Institute of Electrical and Electronics Engineers) and ETSI (European Telecommunications Standards Institute).

1.2.1 Constrained Application Protocol (CoAP)

IETF Constrained RESTful Environments (CoRE) working group has created this protocol. This protocol has designed to be used between devices that are operated on the same constrained network, general nodes on the Internet and between devices that are operated on different constrained networks the only condition is that both networks must joined on the Internet.

This protocol is particularly designed for IoT systems which are based on HTTP protocols. CoAP also makes use of UDP protocol and RESTful architecture. It is used within the social network based applications, mobiles and eliminates redundancy by make use of HTTP get, post, put and delete functions. In the transport layer it uses DTLS for the secure exchange of messages.

1.2.2 MQTT Protocol

Andy Stanford-Clark of IBM and Arlen Nipper of Cirrus has developed MQTT (Message Queue Telemetry Transport) in 1999 [6].

It is mostly used for the purpose of remote monitoring in IoT. This protocol first acquires data from many different devices and transports the data to the IT infrastructure. MQTT protocol makes use of hub-and-spoke architecture. In the hub-and-spoke architecture devices are connected to data concentrator servers. For providing simple and reliable streams of data MQTT protocol works on the top of the TCP.

There are three main components of MQTT protocols subscriber, publisher and broker. The publisher is responsible for generating the data and transfer the information to subscribers through the broker.

1.3 Additional Constraint

The purpose of IoT is to connect each and every machine and device to provide a ubiquitous connectivity. To fulfill this purpose, IoT requires those devices which consume less power, need minimum maintenance and have low computational capabilities [7]. Therefore security solutions should be:

- Light weight
- Need less computation
- Consume less power
- Embedded (RFID's)
- Requires less memory.

2. SECURITY CHALLENGES

Let's consider some of the worst attacks on IoT devices over the last few years and what you can do to prevent falling victim to vulnerabilities [4].

2.1 Attacks on IoT

2.1.1 Stuxnet

This attack had occured between 2010 and 2014. This attack targeted those devices which aren't typical IoT devices according to today's standard that is industrial programmable logic controllers(PLCs). They fall into the category of 'smart controllers'.

The attack was supposedly launched to sabotage the uranium enrichment facility in Natanz, Iran. According to Experts up to 1000 centrifuges destroyed. Although stuxnet was not a typical IoT attack because it depended on the PLC devices to be connected to a machine running the windows operating system.

2.1.2 Mirai Botnet

Plenty of major attacks had occured in 2016. One of those attacks was “Mirai Botnet”. This particular botnet infected numerous IoT devices including older routers and IP Cameras and then used these devices to flood DNS provider Dyn with a DDOS attack.

The Mirai Botnet took down Etsy, GitHub, Netflix, Shopify, SoundCloud, Twitter, Stopify and a number of other major websites [4].

This attack targeted those devices which were running out of date versions of the linux kernel and based on the fact that most users do not change the default username/password on their device.

https://doi.org/10.15623/ijret.2018.0703010 Received: 19-01-2018, Accepted: 02-03-2018, Published: 22-03-2018

2.1.3 Cold in Finland

This attack had occurred in November 2016. This attack was also the DDOS attack. In this attack, the attackers managed to cause the heating controllers to reboot the system continuously so that the heating never actually kicked in.

2.1.4 Brickerbot

This attack worked in same manner as Mirai Botnet, in that it depended upon a DDOS attack and most of the users not changing the default username/password of their device.

One of the major differences between Brickerbot and Mirai Botnet is that Brickerbot simply kills the IoT devices.

2.1.5 The Botnet Barrage

In the case of "the botnet barrage," as the case study dubbed the attack, senior members of the university's IT staff had received complaints of slow and inaccessible network connectivity on campus. Upon examination, the incident commander found that name servers "were producing high-volume alerts and showed an abnormal number of sub-domains related to seafood," [5] according to the preview.

In this attack, the botnet spread from one device to another device via brute forcing default and weak passwords.

2.2 Vulnerabilities Associated with the Attacks

2.2.1 Stuxnet

There are following vulnerabilities associated with Stuxnet

- UPnP
- Username enumeration
- Implicit trust between components
- Firmware extraction
- Root Kit Injection

2.2.2 Mirai botnet

- Known default credentials
- Weak passwords
- Firmware version display and/or last update date
- Missing update mechanism.

2.2.3 Cold in Finland

- Denial of Service
- Sensitive URL disclosure
- Vulnerable UDP Services

2.2.4 Brickerbot

- Denial of Service
- Username enumeration
- Implicit trust between components
- Known default credentials

2.2.5 The Botnet Barrage

- Vulnerable UDP Services
- Denial of Service
- Information disclosure

3. PROBLEM DEFINITION

After looking at these attacks and vulnerabilities associated with these attacks, the problem is very clear that how to eliminate the following vulnerabilities therefore the security of IoT devices can be ensured –

- Universal Plug and Play
- Vulnerable UDP Services
- Firmware Extraction/Manipulation
- Root Kit Injection
- Missing Update Mechanism
- Denial of Service
- Known default credentials
- Sensitive URL disclosure.

4. SOLUTION FOR VULNERABILITIES

This section introduces remediation steps of the problem definition that is discussed above.

4.1 Universal Plug and Play / Vulnerable UDP Services

UPnP is insecure by design. UPnP is a protocol which is designed in such a way that it automatically opens ports in a firewall to allow an outsider to access a hosted server on a local machine that is protected by a firewall.

To prevent SSDP attacks by accessing the networked router's configuration menu with a Web browser, block inbound traffic on UDP port 1900. Disable UPnP altogether to prevent current SSDP and possible future attacks.

Replace and discard any device which has at least one of the following characteristic:

- Any equipment that is shown to have UPnP vulnerabilities where the vendor doesn't plan an update.
- In the case of a router, where UPnP can't be turned off.

4.2 Firmware Extraction/Manipulation

- Ensuring the device has the ability to update and secure update mechanism.
- Ensuring the update file is encrypted using accepted encryption methods and transmitted via an encrypted connection.
- Ensuring the update file does not expose sensitive data and update is signed and verified before allowing the update to be uploaded and applied.

https://doi.org/10.15623/ijret.2018.0703010 Received: 19-01-2018, Accepted: 02-03-2018, Published: 22-03-2018

4.3 Root Kit Injection

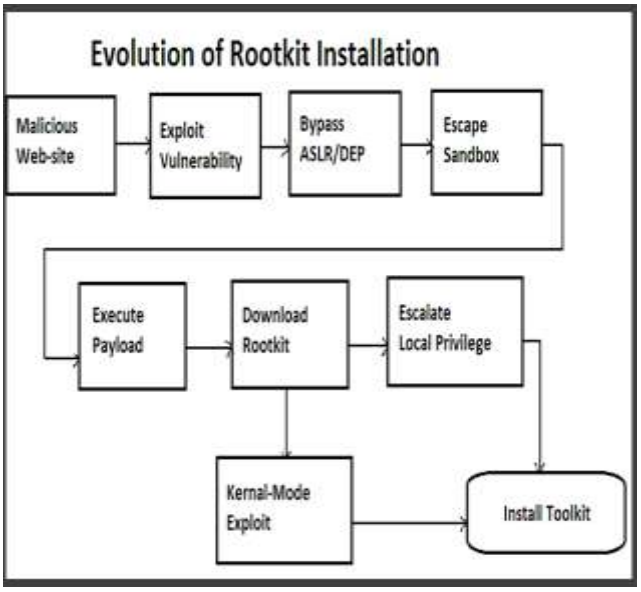


Fig 4.3: Rootkit Installation

- IoT devices running insecure software and services should be patched frequently.
- Restrict access to insecure ports and services.
- Ensure that system has strong administrator password so that attacker cannot take advantage of existing vulnerability.

4.4 Missing Update Mechanism

IoT devices should be capable of establishing secure connection with server, receiving updates and installing updates without much manual intervention.

4.5 Denial of Service

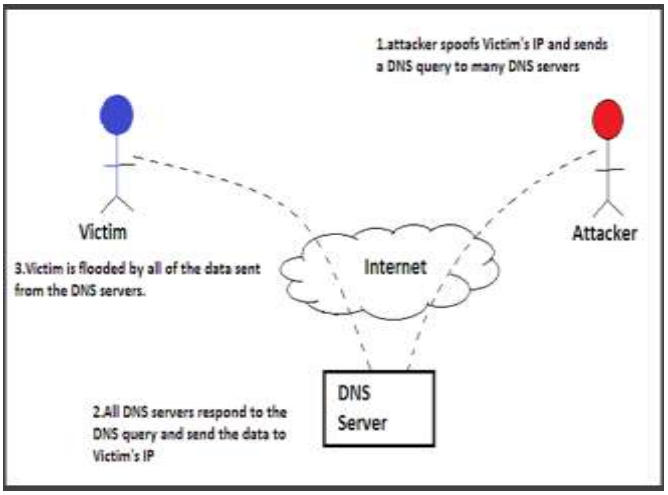


Fig 4.5: Denial of Service

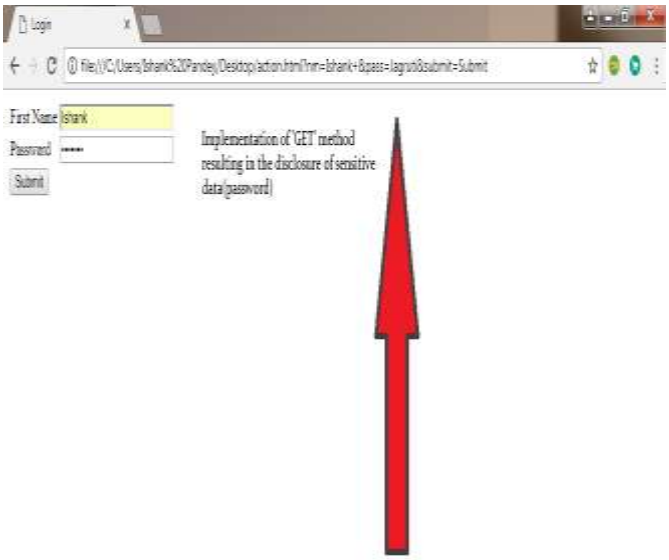
In Today's Internet DoS attacks use packet reflection so that large amount of traffic redirect to a particular entity and hide the origin of their attack.

In order to prevent DoS attack request packets from source should be authenticated first, also multipath communication/hidden network path should be implemented judiciously while designing IoT.

4.6 Known Default Credentials

- IoT application should enforce the user to change the password frequently.
- Username and password must not be hardcoded by the vendors.

4.7 Sensitive URL Disclosure



- IoT application development should follow secure SDLC which includes testing the application at various levels such as to avoid insecure function calls, source code review.
- Sensitive data such as passwords must not be passed through GET method.
- Post method must be used for passing sensitive data because Post requests cannot be cached and they do not remain in user's browsing history which makes the data safe and secure.

5. CONCLUSION

After looking at these attacks, it is very clear that the responsibility for preventing IoT attacks is on both the user and device developer. Every IoT device should deliver with an updated kernel/firmware and all IoT devices must have the ability to regular update as new vulnerabilities are found.

On the other hand, anyone who deploys an IoT device needs to take the time to change the default credentials and constantly be on the lookout for suspicious network activity.

Finally the developers must consider making default credentials change a requirement upon initial deployment of the device.

https://doi.org/10.15623/ijret.2018.0703010 Received: 19-01-2018, Accepted: 02-03-2018, Published: 22-03-2018

Neither Internet of Things is going away nor the attacks on such devices. With just a bit of care during IoT device setup and a constant watchful eye on the network through which IoT device is connected, security breaches can be prevented by way of IoT devices.

REFERENCES

[1] M. Abomhara and G. M. Koien, "Security and privacy in the Internet of Things: Current status and open issues," in Int'l Conference on Privacy and Security in Mobile Systems (PRISMS), 1-8, 2014.
[2] https://www.gartner.com/newsroom/id/3165317.
[3] https://electronicsforu.com/technology-trends/popular-iot-protocols
[4] http://www.zdnet.com/article/5-nightmarish-attacks-that-show-the-risks-of-iot-security/
[5] https://campustechnology.com/articles/2017/02/13/university-hackers-attacked-5000-iot-devices-on-campus.aspx
[6] https://en.wikipedia.org/wiki/MQTT
[7] A Perspective on Available Security Techniques in IoT http://ieeexplore.ieee.org/document/8256859