IOT NETWORK PROTOCOL STACK, CHALLENGES AND SECURITY ISSUES IN DEVELOPING IOT DEVICES

Dr. Vijaya Kumar A V
Assistant Professor, Computer Science and Engineering Department
Proudhadevaraya Institute of Technology, Hosapete
dr.av.vijay@gmail.com, pdit.vijay@gmail.com

Abstract

IoT is a region where digital world converges with physical world. The focus of this paper is on the recommended design considerations for constrained IoT devices with the objective to achieve security by default. The Network Protocol Stack developed by IETF for communication Between IOT devices helps in connecting the smart devices effectively. Since all objects in IoT area unit interconnected through world communication network , security of collected knowledge additionally as communication among devices is crucial facet in IoT. The safety challenges area unit starting from deployment of sensible objects to maintaining users privacy and knowledge privacy. Moreover, confidentiality of communicated info, credibleness of users, integrity of changed knowledge associated access management area unit most evident security necessities in an IoT system. The security challenges are restricting the vast development applications, to advance the development the design consideration are mentioned in this paper. Rather Focusing on the security issues on the communication of the devices, the safety of the IoT devices is also to be monitored.

Keywords: Protocol Stack, Cyebr attack, IoT Design, IoT Router

I. INTRODUCTION

The Internet of Things (IoT) refers to the utilization of showing intelligence connected devices and systems to leverage knowledge gathered by embedded sensors and actuators in machines and different physical objects. IoT is anticipated to unfold quickly over the approaching years and this convergence will unleash a replacement dimension of services that improve the standard of lifetime of shoppers and productivity of enterprises, unlocking a chance that the GSMA refers to because the ‘Connected Life’. IoT products consists of TCP/IP protocol stack and related network protocol stacks. In addition, Option Libraries can be chosen according to the communication function and purpose. More IoT devices make their way into the world, deployed in uncontrolled, complex, and often hostile environments, securing IoT systems presents a number of unique challenges.

II. IoT NETWORK PROTOCOL STACK

The Internet Engineering Task Force (IETF) has developed alternative protocols for communication between IoT devices using IP because IP may be a flexible and reliable standard [1,2]. The web Protocol for Smart Objects (IPSO) Alliance has published various white papers describing alternative protocols and standards for the layers of the IP stack and a further adaptation layer, which is employed for communication [1–4] between smart objects.

(1) Physical and MAC Layer (IEEE 802.15.4) : The IEEE 802.15.4 protocol is meant for enabling communication between compact and cheap low power embedded devices that require an extended battery life. It defines standards and protocols for the physical and link (MAC) layer of the IP stack. It supports low power communication along side low cost and short range communication. within the case of such

resource constrained environments, we'd like alittle frame size, low bandwidth, and low transmit power. Transmission requires little or no power (maximum one milliwatt), which is merely one-hundredth of that utilized in WiFi or cellular networks. This limits the range of communication. due to the limited range, the devices need to operate cooperatively so as to enable multihop routing over longer distances. As a result, the packet size is restricted to 127 bytes only, and therefore the rate of communication is restricted to 250 kbps. The coding scheme in IEEE 802.15.4 has inbuilt redundancy, which makes the communication robust, allows us to detect losses, and enables the retransmission of lost packets. The protocol also supports short 16-bit link addresses to decrease the dimensions of the header, communication overheads, and memory requirements [5]. Readers can ask the survey by Vasseur et al. [4] for more information on different physical and link layer technologies for communication between smart objects.

(2) Adaptation Layer : IPv6 is taken into account the simplest protocol for communication within the IoT domain due to its scalability and stability. Such bulky IP protocols were initially not thought to be suitable for communication in scenarios with low power wireless links like IEEE 802.15.4. 6LoWPAN, an acronym for IPv6 over low power wireless personal area networks, may be a very fashionable standard for wireless communication. It enables communication using IPv6 over the IEEE 802.15.4 [2] protocol. This standard defines an adaptation layer between the 802.15.4 link layer and therefore the transport layer. 6LoWPAN devices can communicate with all other IP based devices on the web. the selection of IPv6 is due to the massive addressing space available in IPv6. 6LoWPAN networks hook up with the web via a gateway (WiFi or Ethernet), which also has protocol support for conversion between IPv4 and IPv6 as today's deployed Internet is usually IPv4. IPv6 headers aren't sufficiently small to suit within the tiny 127 byte MTU of the 802.15.4 standard. Hence, squeezing and fragmenting the packets to hold only the essential information is an optimization that the difference layer performs. Specifically, the difference layer performs the subsequent three optimizations so as to scale back communication overhead:(i)Header compression 6loWPAN defines header compression of IPv6 packets for decreasing the overhead of IPv6. a number of the fields are deleted because they will be derived from link level information or are often shared across packets.(ii)Fragmentation: the minimum MTU size (maximum transmission unit) of IPv6 is 1280 bytes. On the opposite hand, the utmost size of a frame IEEE 802.15.4 is 127 bytes. Therefore, we'd like to fragment the IPv6 packet. this is often done by the difference layer.(iii)Link layer forwarding 6LoWPAN also supports mesh under routing, which is completed at the link layer using link level short addresses rather than within the network layer. This feature are often wont to communicate within a 6LoWPAN network.

(3) Network Layer : The network layer is liable for routing the packets received from the transport layer. The IETF Routing over Low Power and Lossy Networks (ROLL) working party has developed a routing protocol (RPL) for Low Power and Lossy Networks (LLNs) [3]. For such networks, RPL is an open routing protocol, supported distance 7% Plagiarised 93% Unique vectors. It describes how a destination oriented directed acyclic graph (DODAG) is made with the nodes after they exchange distance vectors. a group of constraints and an objective function is employed to create the graph with the simplest path [3]. the target function and constraints may differ with reference to their requirements. for instance, constraints are often to avoid battery powered nodes or to prefer encrypted links. the target function can aim to attenuate the latency or the expected number of packets that require to be sent. The making of this graph starts from the basis node. the basis starts sending messages to neighboring nodes, which then process the message and choose whether to hitch or not depending upon the constraints and therefore the objective function. Subsequently, they forward the message to their neighbors. during this manner, the message travels till the leaf nodes and a graph is made. Now all the nodes within the graph can send packets upwards hop by hop to the basis. we will realize some extent to point routing algorithm as follows.

We send packets to a standard ancestor, from which it travels downwards (towards leaves) to succeed in the destination. To manage the memory requirements of nodes, nodes are classified into storing and nonstoring nodes depending upon their ability to store routing information. When nodes are during a nonstoring mode and a downward path is being constructed, the route information is attached to the incoming message and forwarded further till the basis. the basis receives the entire path within the message and sends a knowledge packet along side the trail message to the destination hop by hop. But there's a trade-off here because nonstoring nodes need more power and bandwidth to send additional route information as they are doing not have the memory to store routing tables.

(4) Transport Layer : Protocol isn't a decent possibility for communication in low power environments because it includes a massive overhead because of the very fact that it's a association oriented protocol. Therefore, UDP is most well-liked as a result of it's a connectionless protocol and has low overhead.

(5) Application Layer : the appliance layer is accountable for formatting and presentation. the appliance layer within the web is usually supported HTTP. However, HTTP isn't appropriate in resource unnatural environments as a result of it's fairly windy in nature and so incurs an outsized parsing overhead. several alternate protocols are developed for IoT environments like CoAP (Constrained Application Protocol) and MQTT (Message Queue mensuration Transport).(a)Constrained Application Protocol: CoAP are often thought of as an alternate to HTTP. it's utilized in most IoT applications [6, 7]. in contrast to HTTP, it incorporates optimizations for unnatural application environments. It uses the EXI (Efficient XML Interchanges) format, that may be a binary format and is much a lot of economical in terms of area as compared to plain text HTML/XML. alternative supported options are inbuilt header compression, resource discovery, auto configuration, asynchronous message exchange, congestion management, and support for multicast messages. There ar four styles of messages in CoAP: non confirmable, empiric, reset (nack), and acknowledgement. For reliable transmission over UDP, empiric messages ar used. The response are often piggybacked within the acknowledgement itself. moreover, it uses DTLS (Datagram Transport Layer Security) for security functions.(b)Message Queue mensuration Transport: MQTT may be a publish/subscribe protocol that runs over protocol. it had been developed by IBM [9] primarily as a client/server protocol. The purchasers ar publishers/subscribers and also the server acts as a broker to that purchasers connect through protocol. purchasers will publish or take a subject. This communication takes place through the broker whose job is to coordinate subscriptions and conjointly certify the shopper for security. MQTT may be a light-weight protocol, that makes it appropriate for IoT applications. however thanks to the very fact that it runs over protocol, it can't be used with every type of IoT applications. Moreover, it uses text for topic names, that will increase its overhead.

MQTT-S/MQTT-SN is Associate in Nursing extension of MQTT [12], that is meant for low power and low value devices. it's supported MQTT however has some optimizations for WSNs. The subject names are replaced by topic IDs, that cut back the overheads of transmission. Topics don't want registration as they're preregistered. Messages are split so solely the mandatory info is shipped. Further, for power conservation, there's Associate in Nursing offline procedure for purchasers UN agency ar in an exceedingly sleep state. Messages are often buffered and later browse by purchasers once they come to life. Purchasers connect with the broker through a entryway device, that resides at intervals the detector network and connects to the broker.

III. CHALLENGES TO BE CONSIDERED WHEN DEVELOPING IOT DEVICES

IoT, along side cloud computing, may be a major contributor to the fourth technological revolution and is inevitably becoming a neighborhood of every of our lives. More and more industries have gradually applied this IoT technology, and an increasing number of enterprises are trying to realize footing within the future IoT world. The challenge with IoT is that a lot of enterprises only specialise in IoT development without evaluating or learning the first challenges that they're facing. Many of those enterprises don't even have any background within the IT industry or software development, but most of them are committed to providing internet-connected devices. Even enterprises that have software and hardware design experience often mistake IoT as other traditional computing technologies and make big mistakes when developing IoT devices. Again and again, facts prove that this practice may be a disaster and can ruin manufacturers' efforts and, ultimately, damage the integrity of IoT. This article will suggests four challenges that each one manufacturers and developers should consider once they plan to enter the IoT industry.

Connectivity : Connectivity is that the first concerning issue, i.e. the way to connect devices to the web and therefore the cloud computing platform. However, to an excellent extent, this is often determined by the device application environment and therefore the sort of communication infrastructure provided to those devices. For example, if you would like to develop a sensible home device, like a web toaster, you'll access a Wi-Fi home router or a ZigBee/Z-Wave IoT router. Therefore, your device must support one or more transmission media. However, in some environments, like the agriculture IoT or smart cars, access to the Wi-Fi network is unavailable, and therefore the mobile network could also be your only choice for connection. Therefore, you want to balance your choice and make design decisions supported possibilities provided by every option and investment. for instance, it's going to be expensive to transmit data through a cellular network to the cloud service, but you'll determine to pick the function first mode or the blockchain mode to create an IoT ecosystem that's less hooked in to cloud computing. Of course, you furthermore may got to know that IoT remains a technology at its early stage and should undergo significant changes or modifications. Too many uncertainties and competition trends exist. Therefore, technologies in use today may become outdated within the future. On the opposite hand, as compared to computers and smartphones which will be quickly replaced by new products, IoT devices have a extended life cycle. for instance, a sensible refrigerator must work for a minimum of five to 10 years. Therefore, you want to develop an idea to make sure that your device can maintain its connectivity and adapt to new technologies when IoT begins to require shape within the future.

Security and Privacy : IoT security has always been a controversial issue. the first challenge to be considered is that security and privacy of IoT are fundamentally different from the network security that we've known. the next lists some key points for security design that are considerable:

1. Physical Security — IoT devices are often located in open fields and are unattended and physically unprotected. you would like to form sure that they are getting to not be maliciously tampered with by a vicious organization, breached by hackers, or operated employing a flat-head screwdriver. Also, you would like to guard data that gets stored on the devices in any form. Although it's costly to embed a security protection component on every IoT device, it's still important to encrypt data on these devices.

2. Security of data Exchange – Data protection is additionally important because data must get transmitted from the IoT sensors and devices to the gateway, then to the cloud. Therefore, use of encrypted transfer protocols could also be a requirement. additionally to encryption, you would like to also consider the authentication and authorization to form sure IoT security. **3. Security of Cloud Storage** — Data stored within the cloud is equally fragile as other parts of the IoT ecosystem. Your platform should be able to protect data stored within the cloud. Protection measures include appropriate encryption, access control, and so on.

4. Update — Security vulnerabilities always exist no matter what proportion effort you pay to strengthen your product code and hardware. during this case, you would like to first have a thought to repair errors and quickly release patches, instead of leaving the errors unfixed for an extended period of some time. Next, you would like to supply customers with an instantaneous and secure method to repair errors. Currently, it's popular to update online devices over the air, but you would like to form sure that the above method itself won't become a security vulnerability.

Regarding privacy, you would like to understand that data collected by IoT devices are easily subject to restrictions on laws and regulations. as an example, a fitness tracker can collect plenty of user information, which is protected by HIPAA within the us. this means if you store this type of knowledge on the cloud server, the data must suits related laws and regulations. As a rule of thumb, you'd better anonymize customer data to avoid storing identity information within the cloud. This rule defends you against legal punishments when incidents occur.

Flexibility and Compatibility Because the IoT pattern is continuously changing, you would like to form sure that your product can support future technologies. However, it requires a balance between software and hardware when designing your product. Developing dedicated hardware for your device helps your device achieve the optimum performance, but also can restrict product update. On the other hand, selecting appropriate storage and computing resources and operating systems (such as Linux, Brillo, or Windows IoT) tailored for IoT may cause degradation of performance, but it allows you to expand your device, use new functions, and fix bugs using patches.

Some vendors may plan to provide appropriate APIs and SDKs whenever possible to allow the developing personnel to feature functions for his or her IoT devices. an honest example is Amazon Echo. This IoT tool can implement the expansion in 1000 different directions using programming. you want to also consider compatibility when designing IoT products. confirm that your IoT device can get seamlessly integrated with users' IoT ecosystem, without increasing complexity or bringing any setbacks to existing experience. For this reason, you'd wish to believe both software and hardware. a perfect situation is that buyers should not be forced to place during a replacement application just because they purchase a replacement smart device for his or her homes. Apple HomeKit and Samsung SmartThings are two typical examples. Both allow the developing personnel to provide new IoT functions for users in environments that users are familiar with. **Data Collection and Processing** Additionally to security and privacy, you would like to also properly plan the thanks to process all collected data. you would like to first evaluate the number of processed and picked up data to manage the size of your cloud storage and meet your platform requirements. what's even more important is how you're going to process the collected data. IoT data is as precious as gold, but it's useless if it gets stored on your server without getting properly processed. Therefore, you would like to seek out out the skills and tools which can best utilize your data. These tools include recruiting data experts and adopting appropriate analysis and machine learning to extract operable insight information from the collected data.

IoT data can complete multiple practical functions, including:

1. Supplement Existing Data — Most enterprises have already got extensive data about their customers before they migrate their services to IoT. Integrating the prevailing data with data collected by IoT devices can bring new business insights and more opportunities for generating revenues.

2. Analyze and Further Divide Users — Data collected by IoT devices can also tell you plenty of data about customers' preferences and characteristics. Analyzing and classifying IoT data can help enterprises better learn their customers' requirements and preferences, and enable them to resolve related problems during a wiser manner.

3. Find Opportunities to reinforce Products — Correct analysis of IoT data helps enterprises determine functions that need to and can not get added to products, and functions that need to be corrected to reinforce the assembly efficiency and ease-of-use. during this manner, enterprises can add appropriate functions to future products and update software accordingly.

IV. SECURITY CHALLENGES

The evolution of IoT will introduce huge number of interconnected devices with which security challenges also will increase considerably [13][14]. Security assurance concerning each and each component of IoT is vital to avoid malicious players in exploiting IoT [15]. Few threats where an attacker can compromise IoT component are as under: -

A. Unauthorized Access to Home or Business Attacker can exploit poor authentication and access control mechanism of devices which control physical access to home or business like electronic doors, locks etc.

B. IoT Botnets As Internet will evolve into IoT with the introduction of the many interconnected devices then it'll also increase the attack surface. Attacker will have access to several smart devices and may cause them to act as botnets by exploiting security weaknesses [16]. for instance, world has seen the sensible manifestation of IoT botnet in September 2016. Cyber security blog called as “Krebs on Security” was hit by largest DDoS attack where 620 Gbps were launched using 1 million IoT devices [17]. ICC2017: WT04-5thIEEE International Workshop on Smart Communication Protocols and Algorithms (SCPA 2017)

C. Monetary Loss Businesses offering various services using IoT are often subjected to monetary loss, if the devices are exploited to not offer intended services.

D. Surveillance Compromised IoT devices during a home or enterprise will enable the attacker to watch and collect valuable data. This illegitimate surveillance may cause harm in terms of privacy loss, loss, theft of property etc.

E. Unauthorized Tracking IoT devices providing location services, if compromised, then can reveal the situation to an attacker.

V. DIFFICULTIES WITH THE SAFETY OF IOT DEVICES

If we were to work out the foremost remarkable weakness within the Internet of Things, it might definitely be safety, not only in consumer devices but also in engineering and manufacturing. IOT questions of safety IoT has got to think beyond usability and specialise in points like: • Software protection. • Implementation of practices against vulnerabilities. • Ensuring the authenticity and integrity of future patches. We now present the ten commonest questions of safety during this domain and their possible solutions.

1.Ecosystem Complexity Since it doesn’t need to appear as if a compendium of stand-alone devices, IoT becomes tangled in its complexity. IoT should to be understood as an upscale, broad and diverse ecosystem that integrates people, communications and interfaces. Although it simplifies life and industrial production, the appliance of the concept isn't simple, as there are many components in its ecosystem. These range from sensors (devices), networks (bridges, routers, WiFi technology, LiFi, etc.) and technological standards (protocols: network, communication and data) and regulations (confidentiality and security).

2. Limited Capacities In Devices This happens with most computers because they are available with limitations in power, processing and memory. As a consequence, they're not managed as advanced security patterns should be, which is why they're at greater risk of being attacked or succumbing to defects. That’s why the architecture of the equipment has got to be scalable because it’s how to supply security.

3. Limited Experience As technologies associated with the web of Things are practically new, we don't have a background of previous threats to allow us to realize failures in protection. There aren't many cyber security experts specializing in IoT. a couple of basic rules are barely available.

4. Threats And Attacks There are computer programs specially designed to attack IoT devices and therefore the ecosystem itself. These are threats called malware. They perform unwanted actions without the user’s consent, causing damage and data theft. Exploit Sequences are other code-based abuses that cash in of fragile points to access the system, hitting the infrastructure with a high to severe impact, counting on the assets affected. Among other threats, we could mention information modification, message reproduction, network failure, system or device failures, data filtering, device modification, etc. Generally, manufacturers shorten the launch time of products, always brooding about the quantity of sales and no end to think about fundamental factors within the design phase, like access control or encryption of data, among many others…

5. Privacy When we accept the contract without reading or understanding the clauses it implies, the privacy of our information is in danger. The number of individuals who click “accept” without understanding or maybe reading the terms when using applications or devices to figure with the web of Things is quite high. Such an action poses a danger. Manufacturers,

wanting to stay one step before the competition, don't care about auditing their equipment sufficiently, and doubtless don't dedicate sufficient resources to make sure that those that bring the devices into their lives are fully confident. A suggestion? Taking advantage of coaching in cyber security or resorting to specialized companies with specific solutions. An example for the primary risks, there's an insecure Smtp freezer wont to send spam; as for the second risk, some devices are so small on support asymmetric encryption.

6. Reduced Costs In order to scale back costs, manufacturing companies could limit safety qualities. The result would be equipment which will never provide adequate protection. we might always be in danger. Reducing costs in hardware also as in development may be a terrible mistake. The user is that the one who finishes up paying completely, considering the clauses that stipulate the businesses in their contracts of terms and conditions.

7. Lack Of Clarity In Responsibilities Regarding safety in IoT devices, there are three key players: manufacturer, service provider and user. within the event of a cyber attack, the assignment of responsibilities isn't entirely clear and may cause conflicts. Another important aspect is how security would be managed when a component is shared between several parts.

8. Lack Of Rigour In processing At the guts of this security problem at IoT is that the user is usually unaware of how the info they transmit via sensor devices are going to be used, because conventional methods of consent are of poor quality, i.e. they are doing not specify the next handling of private information. Such information could reach third parties, and therefore the user won't remember of this diffusion.

9. Safety Versus Efficiency The speed with which IoT devices are to be manufactured limits safeguard considerations, and therefore the budget is probably going to possess an impression, which suggests the corporate would emphasize usability instead of security. In certain occasions, there's no balance to optimize the hardware and requirements of a computer used with the web of Things.

10. Limitation Of Anonymity It's linked to a scarcity of rigour in processing. Sometimes we assume that anonymity is guaranteed in any service we use, but it really isn't. In IoT, to ensure this, it's necessary to optimize the techniques of access control, encryption, design privacy, safeguarding the situation and any basic aspect to avoid any undesired intervention

VI. CONCLUSION

The IoT devices have a greater impact on the human life. The communication between the IoT Smart devices getting expanded the networks. The security concern is the major challenge in developing the IoT devices. The IoT revolution is already well underway, but many organizations are still struggling to implement the safety policies needed to guard themselves from the risks related to these smart devices. By identifying specific vulnerabilities that pose a threat to their networks and educating employees on practicing good habits when it involves their connected devices, companies can take the primary steps toward creating systems that are highly resilient and reduce the risks of knowledge breaches and unauthorized access.

REFERENCES

[1.] Z. Sheng, S. Yang, Y. Yu, A. Vasilakos, J. Mccann, and K. Leung, "A survey on the ietf protocol suite for the internet of things: standards, challenges, and opportunities," IEEE Wireless Communications, vol. 20, no. 6, pp. 91–98, 2013. View at Publisher · View at Google Scholar · View at Scopus

[2.] J. P. Vasseur and A. Dunkels, "Ip for smart objects," White Paper 1, IPSO Alliance, 2008. View at Google Scholar

[3.] D. Culler and S. Chakrabarti, "6lowpan: incorporating IEEE 802.15. 4 into the IP architecture, IPSO Alliance," White Paper, 2009. View at Google Scholar

[4.] J. Vasseur, N. Agarwal, J. Hui, Z. Shelby, P. Bertrand, and C. Chauvenet, "Rpl: the ip routing protocol designed for low power and lossy networks," Internet Protocol for Smart Objects (IPSO) Alliance 36, 2011. View at Google Scholar

[5.] J. P. Vasseur, C. P. Bertrand, B. Aboussouan et al., "A survey of several low power link layers for IP smart objects," White Paper, IPSO Alliance, 2010. View at Google Scholar

[6.] J. W. Hui and D. E. Culler, "Extending IP to low-power, wireless personal area networks," IEEE Internet Computing, vol. 12, no. 4, pp. 37–45, 2008. View at Publisher · View at Google Scholar · View at Scopus

[7.] W. Colitti, K. Steenhaut, N. De Caro, B. Buta, and V. Dobrota, "Evaluation of constrained application protocol for wireless sensor networks," in Proceedings of the 18th IEEE Workshop on Local and Metropolitan Area Networks (LANMAN '11), pp. 1–6, IEEE, Chapel Hill, NC, USA, October 2011. View at Publisher · View at Google Scholar · View at Scopus

[8.] Z. Shelby, K. Hartke, and C. Bormann, "The constrained application protocol (CoAP)," Tech. Rep., IETF, 2014. View at Google Scholar

[9.] B. C. Villaverde, D. Pesch, R. De Paz Alberola, S. Fedor, and M. Boubekeur, "Constrained application protocol for low power embedded networks: a survey," in Proceedings of the 6th International Conference on Innovative Mobile and Internet Services in Ubiquitous Computing (IMIS '12), pp. 702–707, Palermo, Italy, July 2012. View at Publisher · View at Google Scholar · View at Scopus

[10.] D. Locke, "MQ telemetry transport (MQTT) v3. 1 protocol specification," IBM developerWorks Technical Library, 2010, http://www.ibm.com/developerworks/webservices/library/ws-mqtt/index.html.

[11.] U. Hunkeler, H. L. Truong, and A. Stanford-Clark, "MQTT-S—a publish/subscribe protocol for wireless sensor networks," in Proceedings of the 3rd IEEE/Create-Net International Conference on Communication System Software and Middleware (COMSWARE '08), pp. 791–798, Bangalore, India, January 2008. View at Publisher · View at Google Scholar · View at Scopus

[12.] A. Stanford-Clark and H. Linh Truon, "MQTT for sensor networks (MQTT-S) protocol specification," International Business Machines Corporation Version 1, 2008. View at Google Scholar

[13] Saleem, Kashif, Zhiyuan Tan, and William Buchanan. "Security for Cyber-Physical Systems in Healthcare," In Health 4.0: How Virtualization and Big Data are Revolutionizing Healthcare, pp. 233-251, 2017.

[14] Zheng, Guanglou, Rajan Shankaran, Mehmet Orgun, Li Qiao, and Kashif Saleem. "Ideas and Challenges for Securing Wireless Implantable Medical Devices: A Review," IEEE Sensors Journal, 2016.

[15] R.W. Anwar, M. Bakhtiari, A. Zainal, A.H. Abullah, K.N. Qureshi, "Security issues and attacks in wireless sensor network," World Applied Sciences Journal 30.10: 1224-1227, 2014.

[16] IoT Could Become Playground for Botnets Gone Wild, Available: http://www.technewsworld.com/story/83963.html?rss=1.

[17] Source Code for IoT Botnet 'Mirai' Released, Available: https://krebsonsecurity.com/2016/10/source