European Journal of Molecular & Clinical Medicine
ISSN 2515-8260 Volume 07, Issue 10, 2020

An Investigation on Detection of Vulnerabilities in Internet of Things

Abhishek Raghuvanshi[1], Dr. Umesh Kumar Singh[2], Chetan Bulla[3], Dr. Monika Saxena[4], Kishori Abadar[5]

Department of Computer Science & Engineering, Mahakal Institute of Technology Ujjain, India[1]
Director, Institute of Computer Science, Vikram University, Ujjain, India[2]
KLE College of Engineering and Technology Chikodi[3]
Banasthali Vidyapith, Jaipur, Rajasthan 304022[4]
Department of Computer Science, Sadguru Gadage Maharaj College Karad, Maharashtra, India[5]

Abstract:

IoT is a woven mixture of traditional systems, sensors, clouds, mobile applications, Web applications and control systems, affecting every aspect of people's lives. Security concerns are increasing with increasingly heterogeneous devices and data processing. It is also a well known fact that most of the IoT applications and devices are not fully secure and they are vulnerable to certain attacks. On an average, 60 percent IoT applications and devices have some sort of vulnerabilities associated with them. In this research work, an experimental setup is established using server computers, client computers, IoT development boards, sensors, cloud subscriptions. Network host scanning tools and vulnerability scanning tools are used to collect raw data related to IoT based applications and devices. Shodan scanning tool is also used to effectively detect vulnerabilities in IoT devices and perform penetration testing.

Keywords: IoT Security, IoT Privacy, Vulnerability, Shodan Scanner, OWASP

1. Introduction:

IoT has moved to our lives peacefully and steadily over the past decade, advancement in remote correspondence and embedded frameworks and advancement in the vitality of effective radio were the foundational steps in allowing small-minute devices to react and screen and form a world view ready to monitor physical items in another machine management. By linking (Anything) to the two steps that historically existed (if) and (if) that make more applications and administrations that alter the way we deal, the health, the monetary and our public practices, the vision of the IoT empowers the third measuring space[1].

To comprehend IoT security, the threat, the vulnerabilities, and attack must be characterized. A threat is any potential vindictive event that could hurt an advantage. Vulnerability is a shortcoming that makes a danger conceivable. This might be a result of poor plan, setup botches, or improper and uncertain coding procedures.

An attack is an activity that abuses a weakness or authorizes a danger. Instances of assaults incorporate sending vindictive contribution to an application or flooding a system trying to refuse assistance.

The Inner-level security alludes to strategies for ensuring Web applications at the application layer, from pernicious assaults that may uncover private data. For the most part, the web structures are powerless against application level assaults. The principle purpose behind this is website specialists verifiably trust approval rules which are upheld just on the customer side. Moreover, application-layer assaults are appealing to potential assailants, on the grounds that the data they look for eventually lives inside the application itself and it is simple for them to have an effect and arrive at their objectives. Lower layer helps application layer security. The IPsec gives security administrations at the IP layer by empowering a framework to choose the necessary security conventions.

Wikipedia helps the CloudFlare services protect themselves from threats. This technique is effective because CloudFlare has ample experience in the handling of such attacks. For online encyclopedias, this is a really fascinating moment. Spamhous was, for example, secured by CloudFlare's services in March 2013. In addition, in August 2015, the DDoS attack by hijacking unsatisfactory web browsers attacked CloudFlare Client GitHub (an online coding site) [2].

The most damaging one was released on 28 February 2018. Akamai's Prolexic DDoS service has mitigated this threat. Akamai has invested in high-DDoS defense. It is made up of seven scrubbing centers and 150 personnel assigned to tackle DDoS assaults. It is also clear that it takes huge sums of investment in money, resources and time. Although such attacks remain vulnerable to a large number of memcached servers (approx. 50 K) [3].

The DDoS attack on Botnets in October 2016 infected a significant number of IoT-based devices [4]. Few standard DDoS attacks threaten the railway transport networks. The DDoS attacks hit the rail network in Sweden in October 2017, which delayed the service, collapsed the IT system that monitors the location of the trains, and disassembled the corresponding email networks, websites and traffic maps. IoT security is therefore the hour-need for today's network media to provide safe and streamlined services in an IoT environment.

2. Related Work:

Z. Uh, Li et. A approach is proposed by al. [5]. To find multiple bugs in the method, this approach makes use of code inspection. It is asserted that all the vulnerabilities listed in NVD can be detected by the proposed procedure. Hey, Uwagbole et. Al. [6] prepared attack data sets. Classification on data sets is then applied. This classifier assists in vulnerability identification.

And Guojun et. Al. [7] implemented a web crawler. This web crawler makes use of clustering of documents. It is TF-IDF-based. From Medeiros et. Al. [8] introduced a system of evaluating source code. The data mining concepts are based on this methodology. New procedures for differentiating web server flaws have been developed by Adnan Masood and Jim Java [9].

A new approach has been implemented by Iberia Medeiros and Nuno Neves[10] to find bugs in webapps. It also makes use of source code data mining and static investigation. In all web apps, Marcelo Invert

Palma Salas, Paulo Licio deGeus and Eliane Martins[11] found that XML injection is a significant weakness. Nearly all developed webapps are still found to be suffering from XML injection problems.

Madan et al[12] performed an investigation into various international standards such as ISO-27002, OWASP, COBIT, and PCI/DSS, which demonstrates the degree of inclusion of countermeasures that rely on the protection of web applications from the point of view of forestalling web application attacks primarily from code infusions. In basically all the international standard rules, the developers conveyed that the concept of acceptance is firmly defined and broadly veiled, but assaults are on the rise because of the vulnerabilities of the Code infusion. In order to restrict the safety gauges, there is a dire need to make engineers and consumers aware of the safety gauges and to encourage them to carefully apply the specifications.

Teodoro and Serrao[13] spoke of the immediate consequences of the lack of protection and the role of value in the life cycle of product improvement, and of the key factors concerning them. In addition, a lot of security mechanized tools and techniques have been proposed by the developers that can be used throughout the SDLC as a way to enhance the security and quality of simple electronic applications. They also requested that every association for network enhancement should provide planning and understanding, prioritization of web application, risk classification, specification of security specifications, threat modeling, audits of architecture configuration, safe coding and post-sending security assessment.

A methodology was proposed for the counteraction of Denial of administration to utilize the site diagram structure, to relieve flooding assaults on a site, utilizing the new Web Referral Architecture for Privileged Service ("WRAPS") by Wang and Reiter [14]. It permits a genuine customer to acquire a benefit URL through a straightforward snap on a referral hyperlink, from a site trusted by the objective site. Utilizing that URL, the customer can get favored access to the objective site in a way, that is far less defenseless against a Distributed Denial of-Service (DDoS) flooding assault than ordinary access would be. The proposed model doesn't expect changes to the web customer programming, and is amazingly lightweight for referrer sites, which makes its organization simple. The creator introduced the plan of WRAPS, and the execution of a model framework. This model exhibits that WRAPS empowers real customers to interface with a site easily regardless of an exceptionally concentrated flooding assault, at the expense of little overheads on the site's ISP's edge switches. The vast majority of the sensible assaults in web applications traded off their arrangements and practices. Assaults against strategies and methods come in numerous pretenses. They likewise show themselves outside of Web applications. Assaults against business rationale can hurt Web locales; however aggressors can likewise utilize Web destinations as the middle person.

Rather than the database driver in the SDriver, an intermediary was planned by Liu et al [15], the called SQLProb (SQL Proxy-based Blocker), which could obstruct the SQL infusion in web applications, by setting an intermediary blocker in the framework. The SQLProb extricates client contribution from the

application generated inquiry, in any event, when the client input information has been inserted into the question, and approves it with regards to the created inquiry's syntactic structure. It approves client contributions by extricating and adjusting them against substantial contributions, by utilizing and improving the hereditary calculation. The SQLProb is a finished discovery approach that doesn't require adjusting the application or database code, in this manner maintaining a strategic distance from the multifaceted nature of polluting, learning, or code instrumentation. What's more, the information approval procedure doesn't require metadata or learning. The SQLProb is autonomous of the programming language utilized in the web application. Be that as it may, the impediment of the framework is the mix of the intermediary framework, which will be the overhead for the web application to forestall the SQL infusion. Likewise, it doesn't bolster the outlandish questions which are linguistically right yet lead to SQL infusion.

Gruschka et al [16] presented a far reaching stream-based WS-security handling framework that gives an increasingly effective preparing in administration processing and improves the strength against various kinds of Denial-Of-Service (DoS) assaults. Their motor is fit for handling standard utilization of WS-Security in a spilling approach. Their framework was intended to deal with, e.g., any course of action, number, and settling level of mark and encryption systems, shutting the hole toward increasingly productive and reliable Web Services.

Ladan [17] grouped Web Services measurements into two principle classifications as auxiliary measurements, and quality measurements. The creator has overviewed the majority of the current Web Services measurements which are found in the writing. Most of the measurements fall throughout the below average which incorporate execution, unwavering quality, adaptability, limit, vigor, exemption taking care of, exactness, uprightness, openness, accessibility, interoperability and security.

Hoquea et al [18] contemplated the conduct and conceivable effect or seriousness of harms. At that point, the creator sorted the assaults into various unmistakable classes. They gave scientific classification of assault apparatuses in a steady manner to help organize security specialists. They introduced a wideranging and arranged review of existing devices and frameworks that can bolster the two assailants and system protectors. The creators have given a conversation on the benefits and negative marks of such apparatuses and frameworks for better comprehension of their abilities.

Binbin Qu et al [19] clarified the plan of a model framework against SQL infusion and cross-site scripting vulnerabilities. The primary strides of the location are different into building the pollute reliance chart for the program by the static examination of source code. A limited state automaton is utilized by them to speak to the estimation of polluted string and confirms whether the program has powerful safe taking care of for the client contribution by coordinating with the assault design. They executed the model framework for programmed recognition dependent on spoil reliance examination.

3. Detection of Vulnerabilities in IoT Devices and Applications:

This experimental setup is established in laboratory of Mahakal Institute of Technology, Ujjain (India) using server computers, client computers, IoT development boards, sensors, cloud subscriptions etc. Network host scanning tools and vulnerability scanning tools are used to collect raw data related to IoT based smart cities.

First, we have developed three IoT based projects in our laboratory, namely:

- IoT Based Weather Monitoring System
- IoT Based Smart Irrigation system
- IoT Based Automated Street Lighting

Following hardware were used to develop IoT Based System:

- Arduino Uno
- ESP8266
- DHT11 Sensor
- Breadboard & Wire
- Pin Connectors

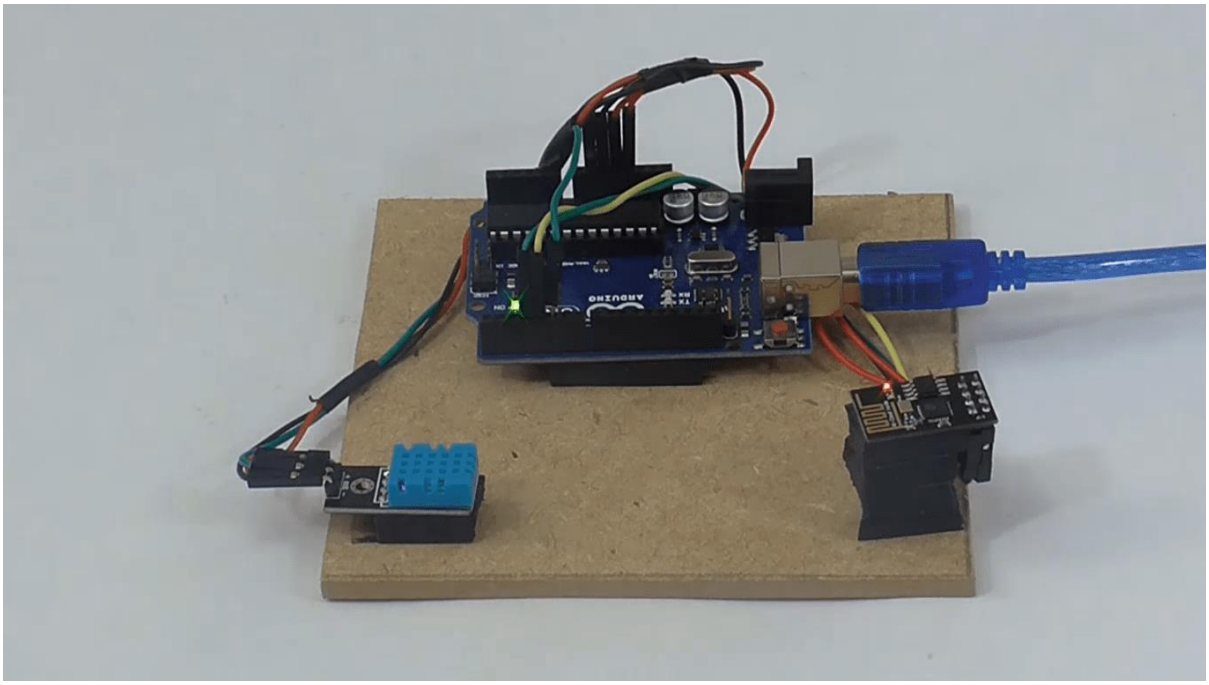

Figure 1: IoT Based Weather Monitoring System

The DHT11 sensor collects data and then this data is sent to ThingSpeak cloud by using ESP8266 communication module. ThingSpeak is an IoT analytics software service for aggregating, visualizing and analyzing live cloud data feeds. From your smartphones you can send ThingSpeak data, generate live data instantly and send warnings.

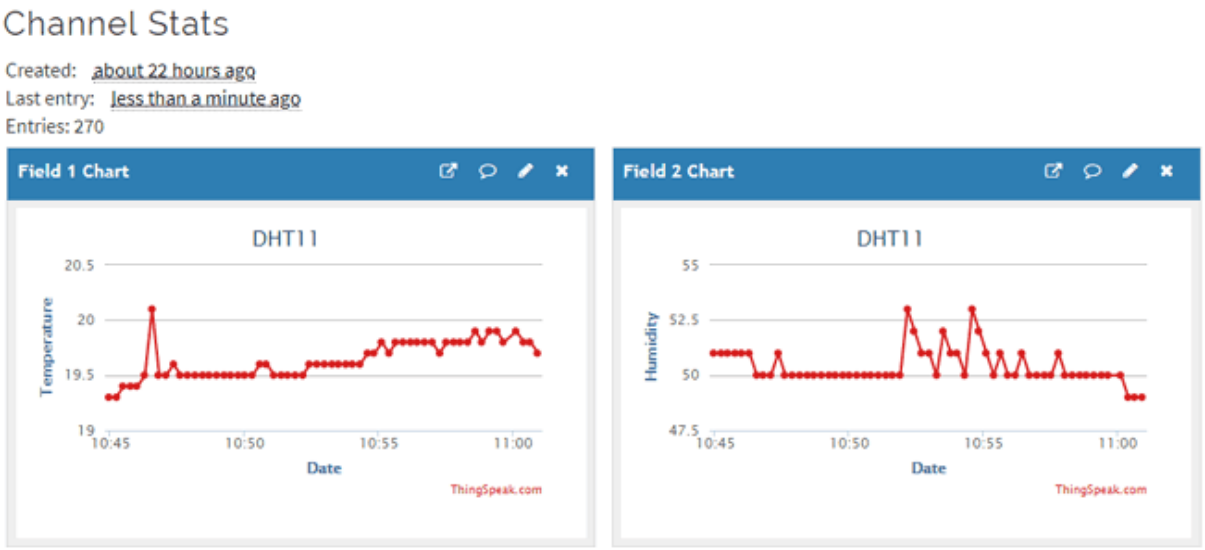

Figure 2 : Thing Speak Cloud Analysis

After experimental setup, shodan [20] scanner is to perform an investigation of various security flaws in IoT network. Launched in 2013, Shodan is a worldwide search engine for IoT devices (Internet of Things). Webcams, protection networks and routers are just a few of the gadgets which can glimpse our lives behind closed doors until they are linked to the Web, should poor safety become the key.

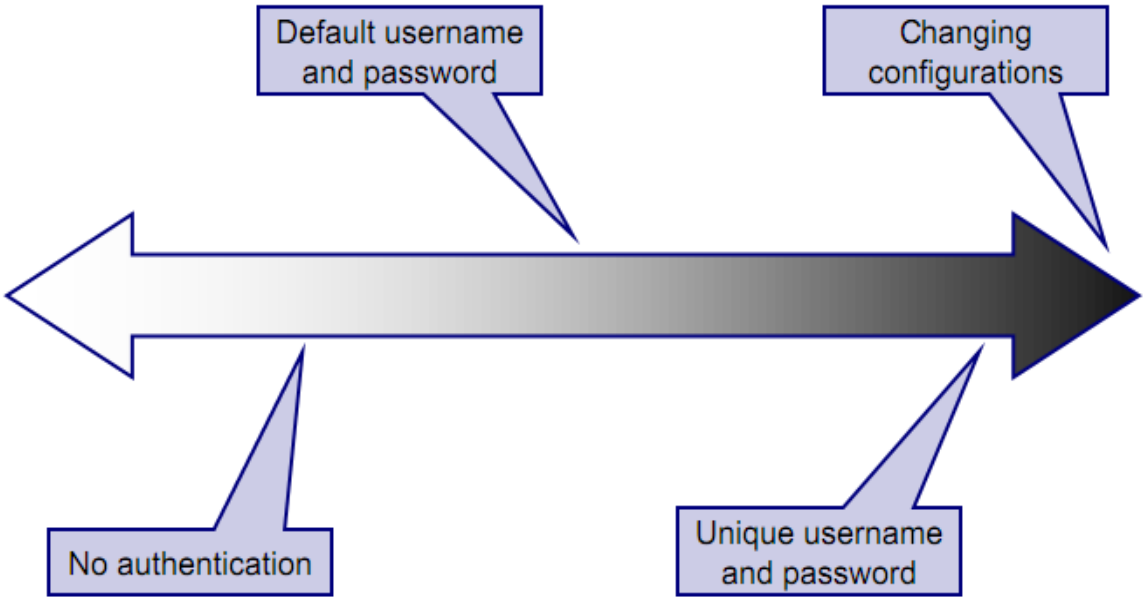

Figure 3: Probable search in IoT using Shodan

Figure 3 shows most common searches in shodan scanner. We used following filters and commands in Shodan

- city: find devices in a particular city
- country: find devices in a particular country
- geo: you can pass it coordinates
- hostname: find values that match the hostname
- net: search based on an IP or /x CIDR
- os: search based on operating system
- port: find particular ports that are open
- before/after: find results within a timeframe
- Find Apache servers in Mumbai:

Figure 4 and 5 below shows scanning particular port and IP address by using shodan**.** Shodan enables the identification of the locations of certain computers and their owners, who are connected to the Internet at any given time. This sort of computer could be in almost any system, including company networks, tracking cameras, ICS and intelligent homes. Shodan tries, by collecting data via associated server ports, to take up the device banner directly. Banner selection helps to find compromised networks is the main move for penetration checks. In the vulnerability area of the search engine Shodan also checks for relevant vulnerabilities.

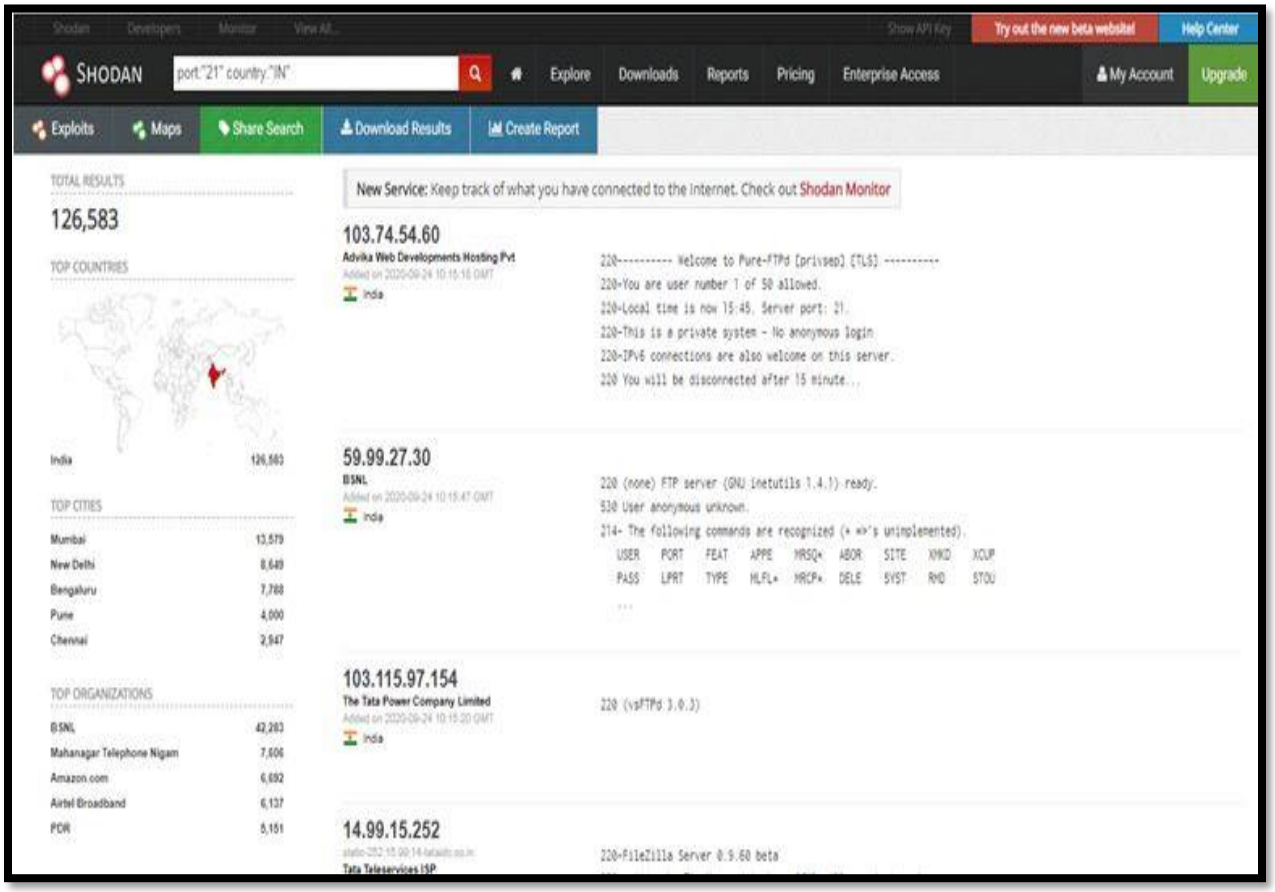

Figure 4: Scanning a particular port by using Shodan

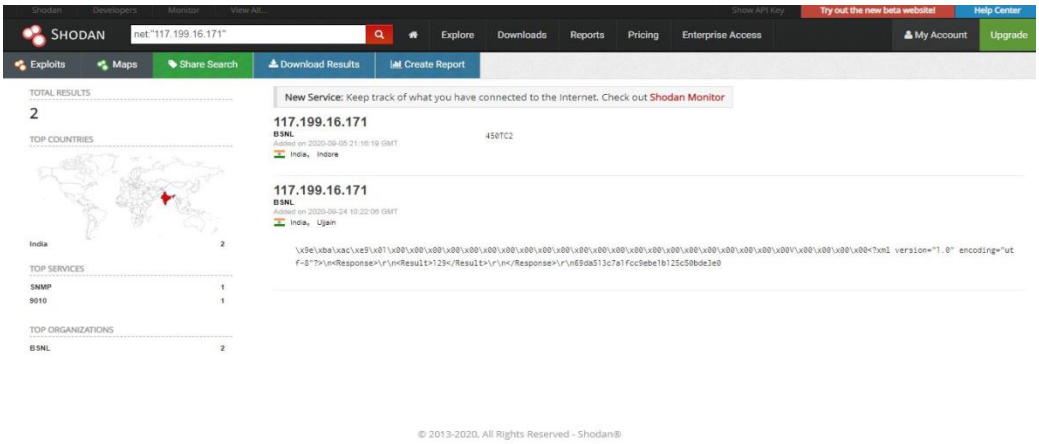

Figure 5: scan a particular IP address by using shodan

Banner is the simple data device that Shodan receives and what you can look for. Shodan can only check for the data property by example. Depending on service form, the quality of the data property can differ greatly. For instance, in figure 6 here is a popular HTTP banner:

HTTP/1.1 200 OK
Server: nginx/1.1.19
Date: Sat, 03 Oct 2015 06:09:24 GMT
Content-Type: text/html; charset=utf-8
Content-Length: 6466
Connection: keep-alive
The above banner shows that the device is running the **nginx** web server software with a version of **1.1.19**.

Figure 6: A Typical Banner in Shodan

Using SHODAN for penetration testing requires some basic knowledge of banners including HTTP status codes.

Status Code	Description
200 OK	Request succeeded
401 Unauthorized	Request requires authentication
403 Forbidden	Request is denied regardless of authentication

Table 1: HTTP code

From above table 1, it is clear that when HTTP status code is 200 OK then no authentication is required and banner is loaded. After Shodan scanning and analyzing OWASP[21] (Open Web Application Security Project, the top most vulnerabilities in IoT are:

- Weak, guessable, or hardcoded passwords
- Insecure network services
- Insecure ecosystem interfaces
- Lack of secure update mechanism
- Use of insecure or outdated components
- Insufficient privacy protection
- Insecure data transfer and storage
- Lack of device management
- Insecure default settings
- Lack of physical hardening

4. Conclusion:

IoT has become the most crucial component in modern world between different types of services and clients. IoT platform is used in many real world applications ranging from healthcare to government administration. It is also a well-known fact that most IoT applications are not fully secure, and are vulnerable to certain attacks. IoT application protection is a key area of concern. This paper provides a study of the approaches used to detect vulnerabilities in IoT devices and applications. An experimental set up is established by implementing IoT applications using arduino uno and sensors. Data is acquired into the cloud to perform simulations. Shodan scanner is used to detect various vulnerabilities in IoT devices and applications like- week password, web camera vulnerabilities etc. This will help future researchers to make solutions to mitigate vulnerabilities and also develop an Intrusion Detection System to enhance security features in IoT ecosystem.

References:

[1] Raghuvanshi, A., & Singh, U. (2020). Internet of Things for smart cities- security issues and challenges. *Materials Today: Proceedings*. doi: 10.1016/j.matpr.2020.10.849

[2] Dunn JE. Wikipedia fights off huge DDoS attack; Sep 11, 2019. https://nakedsecurity.sophos.com/2019/09/11/wikipedia-fights-off-huge-ddos- attack/. Accessed September 18, 2019.

[3] World's largest DDoS attack: US firm suffers 1.7 Tbps of DDoS attack; March 6, 2018. https://www.hackread.com/worlds-largest-ddos-attack- us-firm-suffers-1-7-tbps-of-ddos-attack/. Accessed January 8, 2019.

[4] Osborne C. GitHub suffers "largest DDoS" attack in site's history; March 30, 2015. https://www.zdnet.com/article/github-suffers-largest-ddos- attack-in-sites-history/. Accessed January 8, 2019.

[5] Z. Li, et Al., "VulPecker: an automated vulnerability detection system based on code similarity analysis", ACM, Proc. of the 32 Annual Conference on Computer Security Applications, pp. 201213, 2016.

[6] S. O. Uwagbole, W. J. Buchanan, & L. Fan, "Applied machine learning predictive analytics to SQL injection attack detection and prevention", IEEE, Symposium on Integrated Network and Service Management (IM), 2017 IFIP/IEEE, pp. 1087-1090, 2017.

[7] Z. Guojun, et. Al., "Design and application of intelligent dynamic crawler for web data mining" IEEE, In Automation (YAC), 2017 32nd Youth Academic Annual Conference of Chinese Association, pp. 1098-1105, 2017.

[8]I. Medeiros, N. Neves, & M. Correia, "Detecting and removing web application vulnerabilities with static analysis and data mining", IEEE, IEEE Transactions on Reliability, Vol 65, Issue 1, pp. 54-69, 2016.

[9] Adnan Masood, Jim Java, "Static Analysis for Web Service Security – Tools & Techniques for a Secure Development Life Cycle", International Symposium on Technologies for Homeland Security, pp. 1-6, 2015.

[10] Ibéria Medeiros, Nuno Neves, "Detecting and Removing Web Application Vulnerabilities with Static Analysis and Data Mining", IEEE TRANSACTIONS ONRELIABILITY, pp.1-16, 2015.

[11] Marcelo Invert Palma Salas, Paulo Lício de Geus, Eliane Martins, "Security Testing Methodology forEvaluation of Web Services Robustness - Case: XMLInjection", IEEE World Congress on Services, pp. 303-310, 2015.

[12] Madan, S. "Security Standards Perspective to Fortify Web Database Applications from Code Injection Attacks", International Conference on Intelligent Systems, Modelling and Simulation, pp.226-233, 2010.

[13] Teodoro, N. and Serrao, C. "Web application security: Improving critical web - based applications quality through in - depth security analysis", In International Conference on Information Society (i-Society), pp.457-462, 2011.

[14] Wang, X., and Reiter, M. K. "Using Web-Referral Architectures to Mitigate Denial-of-Service Threats", Journal IEEE Transactions on Dependable and Secure Computing, Vol.7, No.2, pp.203-216, 2010.

[15] Liu, A., Yuan, Y., Wijesekera, D. and Stavrou, A. "SQLProb: a proxybased architecture towards preventing SQL injection attacks", In proceedings ACM Symposium on Applied Computing (SAC'09), pp.2054-2061, 2009.

[16] Gruschka, N, Jensen, M, Lo Iacono, L & Luttenberger, 'Server-side Streaming Processing of WS-Security', IEEE Transactions on Services Computing, vol. 4, no. 4, pp. 272-285, 2011.

[17] Ladan, MI, 'Web Services Metrics: A Survey and A Classification', Journal of Communication and Computer, vol. 9, no.7. pp. 824-829, 2012.

[18] Hoque, N, Monowar, H, Bhuyan, Baishya, RC, Bhattacharyya, DK & Kalita, 'Network Attacks: Taxonomy, tools and systems', Journal of Computer and Network Applications (Accepted), Available from: <http://dx.doi.org/10.1016/j.jnca.2013.08.001>.4 October 2013.

[19] Binbin Qu, Beihai Liang, Sheng Jiang & Chutian Ye, 'Design of Automatic Vulnerability Detection System for Web Application Program', Proceeding of Fourth IEEE International Conference on Software Engineering and Service Science (ICSESS), pp. 89-92, 2013.

[20] Shodan. (2021). Retrieved 29 January 2021, from https://www.shodan.io/

[21] OWASP Top Ten Web Application Security Risks | OWASP. (2021). Retrieved 29 January 2021, from https://owasp.org/www-project-top-ten/