Discovering and Investigating Vulnerabilities in Smart-Home IoT Devices Using Shodan

Sruti Bhagavatula, Vidya Gopalakrishnan, Vibha Iyengar, Alin Nagraj

ABSTRACT

The number of Internet of Things (IoT) devices being used in the wild has been increasing at an alarmingly high rate in the past few years. Given the range of recent cyber attacks like the Mirai botnet exploiting vulnerabilities in vulnerable webcams, understanding how IoT devices can be compromised and how to protect them from such attacks is an important consideration. We utilize the Shodan search engine to extract different classes of IoT devices visible to the Internet and to understand how attackers can find and exploit them. In our project, we focus on smart-home devices and their characteristics and vulnerabilities through analyzing Shodan search results. Smart-home IoT devices could possess vulnerabilities due to the use of vulnerable protocols or due to missing or weak authentication mechanisms. Previous work in this space includes quantitative assessments and evaluations of attack surfaces on Industrial Control Systems (ICS) and Supervisory Control And Data Acquisition (SCADA) devices. However, to the best of our knowledge, there is no existing study which links results from Shodan to study the extent of vulnerabilities in the protocols that these open Smart-home devices use. Therefore, we believe that our measurement study is novel, and can be used in further research in the field of Smart homes and MQTT protocol (Message Queue Transport Telemetry Protocol).

1. INTRODUCTION

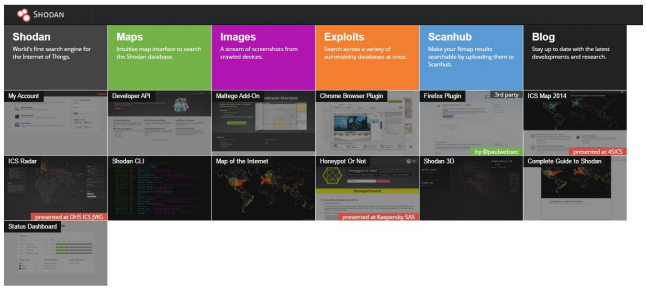

Figure 1: Shodan today provides much more than what it started out as

Shodan is a search engine for Internet of Things (IoT) devices that are visible and connected to the open Internet [12]. This service was initially intended to look up publicly accessible devices, concentrating on SCADA devices specially, but with the rise of various classes of IoT devices, SCADA devices have become only one small part of the search space on Shodan. The search engine has grown in functionality since its creation and is used for a multitude of purposes.

While this search engine was originally intended for non-malicious use and research purposes, the information extracted from Shodan can be used to discover and attack various vulnerable devices. This has been proved by the previous work in this space which includes quantitative assessments and evaluations of attack surfaces on Industrial Control Systems (ICS) and Supervisory Control And Data Acquisition (SCADA) devices. With IoT devices now spanning more and more of the globe, entering every household and office, it is important to measure the increase in attack surfaces due to these devices. The emergence of these devices has also led to the introduction of several application-level light weight protocols like MQTT, Zigbee, Zwave, and BLE (Bluetooth Low Energy).

According to IHS Markit, 80 million smart home devices were delivered worldwide in 2016, which is 64 percent more than that in 2015. Most of these devices use no authentication or they use default passwords. The Mirai botnet, for example used around 68 pairs of default usernames and passwords to capture IoT devices and form a giant botnet. Multiple major DDoS attacks on DNS services of the DNS service provider, DYN, occurred in October 2016, resulting in the inaccessibility of several high-profile websites such as GitHub, Twitter, Reddit, Netflix, Airbnb and many others [1]. The Mirai malware that was installed on many IoT devices (webcams) was the reason for the release of more than 1 TB data per second, which led to the DDoS attack.

In this study, we focus on smart-home IoT devices such as thermostats, smart-garage doors, lights, webcams, home surveillance devices, and many more, which are visible through the Shodan search engine.

1.1 MQTT: Message Queue Telemetry Transport

An important facet of smart-home devices is the set of protocols used by them, One such protocol called the Message Queuing Teemetry Transport (MQTT) works on the publisher/subscriber model . MQTT consists of asynchronous message communication between a broker and a client, transmitted by verifying against a list of topics [2]. A client can either be a publisher or a subscriber or both. A publisher posts content about a specific topic it advertises. A subscriber receives content about a topic it subscribes to. An MQTT broker is a server that implements the MQTT protocol. It mediates communication between MQTT subscribers and

Table 1: Connection codes from MQTT Brokers

Return Code	Return Code Response
0	Connection Accepted
1	Connection Refused, unacceptable protocol version
2	Connection Refused, identifier rejected
3	Connection Refused, Server unavailable
4	Connection Refused, bad user name or password
5	Connection Refused, not authorized

publishers. MQTT defines three quality of service (QoS) levels for message delivery, with each level designating a higher level of effort by the server to ensure that the message gets delivered. Higher QoS levels ensure more reliable message delivery but might consume more network bandwidth or subject the message to delays due to issues such as latency. MQTT acknowledges connections using connection codes/return codes. Table 1 contains the list of connection codes supported and returned by an MQTT broker.

Figure 2 shows the threat model of the MQTT protocol. The MQTT protocol has vulnerabilities that can be exploited if the broker IP address is known. Shodan provides IP address and topics lists, making attacks more possible. We describe four possible vulnerabilities that can be exploited in the MQTT protocol:

1. **Addition of new subscribers without authentication:** The MQTT broker maintains a dynamic list of subscribers, wherein for vulnerable brokers, anyone with the known list of topics served by the broker, can be a subscriber. This means that an attacker can connect to the nearest MQTT broker and fetch content meant to be served to only relevant clients.

2. **No hard enforcement of authentication:** The MQTT protocol does facilitate a way to authenticate the clients connected to the broker by maintaining a configuration file named, Mosquitto.conf. This configuration file maintains a list of authorized user names and passwords and is referenced if the broker sets the “Allow-anonymous" field within the MQTT data-payload to “false". Although this capability is present, it is not utilized by most broker vendors.

3. **Unlimited topic subscriptions:** There is no upper limit on the number of topics a broker can maintain. This means that an attacker can use a single client to publish to a huge list of topics and also subscribe to the same topics with multiple clients. This would lead to excessive bandwidth utilization within the network of IoT devices, thus adversely affecting the smooth communication between legitimate IoT devices.

4. **Wild Cards:** The existence of wild-cards during topic subscription encompasses the above 3 vulnerabilities. An attacker can fetch all the topics served by the broker by initiating a subscription with a wild-card topic. Further, this could be used to create a botnet using these brokers by talking to devices using wild-card topics.

The main objectives of our project are the following:

- **Assessment of the smart-home IoT attack surface:** This involves discovering vulnerable smart-home devices visible to the public Internet through Shodan

- **Measuring characteristics across discovered vulnerable devices:** This involves visualizing the geographical distribution of devices: characterizing the nature of the IP address owner; and recording the different protocols these devices use.

- **Measuring prevalence of devices using vulnerable protocols:** This involves measuring the amount of devices that do not require authorization or authentication within their protocol; and studying vulnerabilities in the MQTT protocol in specific.

2. RELATED WORK

There has been a large body of research concerning vulnerability analysis and assessment in IoT and IIoT (Industrial Internet of Things) systems. Industrial Control Systems (ICS) are the most widely analyzed devices on Shodan [3, 4, 10]. This is mainly because by design they are not supposed to be open to the Internet. Additionally, a sizable amount of ICS and SCADA devices were found to be easy to look up and frequently with no security measures in place [9].

Leverett measures vulnerabilities in ICS devices using Shodan which provides a good guideline for our measurements of smart-home devices. While our measurement methodology differs from those for ICS devices, it gives a baseline of considerations to be made when doing this sort of measurement study [8].

With the success of the Mirai botnet, it has become important to secure all small-scale IoT devices such as vulnerable surveillance cameras, smart bulbs, web cameras, smart garage-doors, and other smart devices used in the home today. Williams describes the potential outcomes of this issue in a white paper. [5, 11].

In the realm of MQTT, Hunkeler et al. describe an enhanced lightweight MQTT protocol that can be run on low-battery in wireless sensor networks [6]. M.Singh et al proposed a secure version of MQTT in 2015 that they call SMQTT, which was based on Key/Cipher text Policy-Attribute Based Encryption(KP/CP-ABE) using lightweight Elliptic Curve Cryptography [13]. This address the authentication problem described earlier. Some studies also focused on the Transport Layer Security (TLS) client authentication, by imposing hardware security elements, while dealing with IoT MQTT communication [7].

3. METHODOLOGY

To carry out our analysis, we performed the following steps:

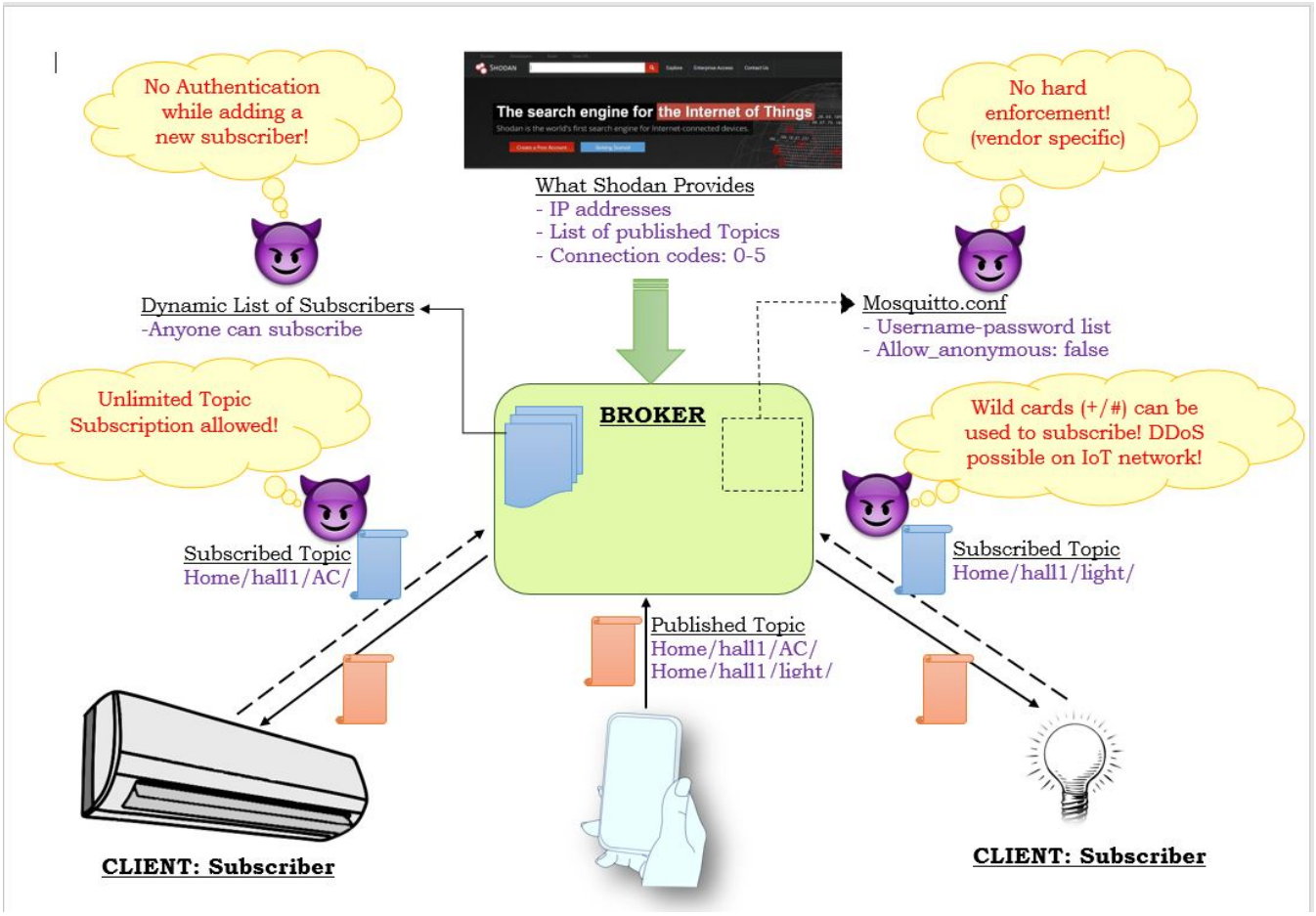

Figure 2: Threat Model of the MQTT Protocol. The Shodan search engine provides IPs of the MQTT brokers and the topics they publish, which allows to exploit the protocol implementation vulnerabilities.

1. **Querying search terms:** Populating appropriate search queries for which querying will return interesting results
2. **Result processing:** From the set of results shown, extracting only the results that are actually devices whose interface is visible, and removing false positives of non-device results from Shodan
3. **Banner Analysis:** Analyze the Shodan result headers or “banners" for the following information:
    - characteristics such as geolocation, country, HTTP response codes when connecting, and owner of the IP address/ISP
    - payload data for devices using the MQTT protocol to quantify how many devices require no authorization

Throughout our study, we only look at merely the results that come from Shodan and do not attempt any unauthorized accesses/logins to exposed devices. For instance, as soon as a prompt for login/password appears by visiting any of the returned result addresses, we immediately terminate the connection and do not proceed any further to guess/brute-force the credentials for any of these devices.

We now describe in detail the steps of our methodology:

3.1 Querying Search Terms

We constructed a set of search-keywords to search for on Shodan with the goal that they return results relevant to smart-home devices. Table 2 contains a list of all search query-terms we search for on Shodan. Although this is not an exhaustive list of search keywords, we believe that it is a representative list of the possible devices we could find commonly in a Smart-home environment. All search keywords fall into one of the two categories described below.

- **Category 1**: General names of smart-home devices e.g., bulbs, thermostat, refrigerator, home-security, surveillance, etc.
- **Category 2**: IoT specific protocols and standards e.g., zigbee, zwave, ble, mqtt, etc.

3.2 Result Filtering

Table 2: Search Keywords for Category 1: Smart-home device list

Search Term	Description
"thermostat"	Simply the term "thermostat" to yield results of potential smart-thermostats connected to the internet
"light","bulb", "light+bulb"	These terms would yield results of potential smart-bulbs connected to the internet
"garage+door"	Potential smart garage-door openers that are accessible over the Internet
"microwave"	Smart-microwaves
"home+surveillance", "webcams"	Potential surveillance systems and webcameras that are still out in the Internet
"MQTT", "zigbee", "zwave", "ble"	IoT specific communication protocols majorly used for home automation

Shodan crawls the Internet to obtain banner and version information for IP addresses that offer HTTP, TELNET, FTP and HTTPS/SSL services. The results that were obtained using Shodan were not all Internet-connected device interfaces. Results were a mixture of websites that contained the search word in the page text as well as actual devices whose interfaces were visible and controllable by visiting that page. In our analysis, we aimed to look for Shodan search results that would imply an actual device, hence, this step involved manual filtering of the returned results for actual devices.

In order to do this filtering for the large datasets Shodan returned, we first obtained all search results and manually inspected the web-interfaces/URL redirections for each of the resulting IPs. We followed the following steps for this filtering process: The following procedure was undertaken:

1. We collected the entire result dump for each search keyword using the Python API for Shodan
2. From the result dump, we manually flagged just the true-positive results by visiting and inspecting the web interface or domain that the resulting IP connected to. This set of true positives is the dataset on which we perform all subsequent analyses.

   Figure 3 is an example of a true positive result representing a real device interface. Figure 4 is an example of a false positive result that is merely a website which we removed from our dataset.

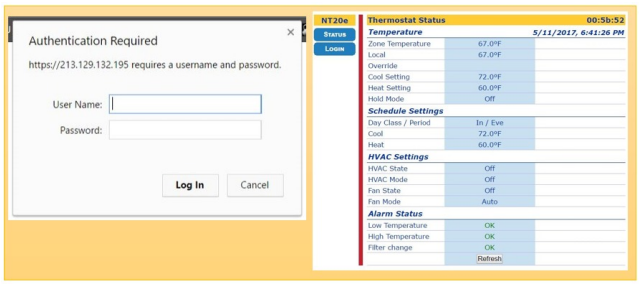

Figure 3: An example of a true positive result from the returned results. On the right is a thermostat's web interface and on the left is a login screen to a garage-door

3.3 Analysis of the Shodan Banner Data for Filtered Results

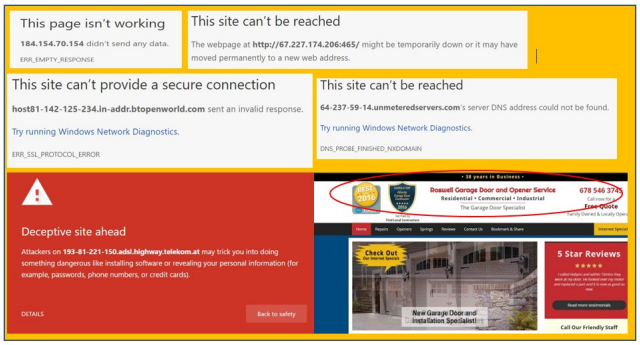

Figure 4: An example of a false positive result from the returned results. These results are web-pages and warning pages that are not direct smart-home device interface

1. **Measurement of Characteristics**: We first aimed to obtain a geographic footprint of all visible devices and collected information about the general geographic locations of the devices as well as more fine-grained location information. We report Table 3. We additionally record information such as the HTTP response code returned when Shodan accesses a device interface, the ISP, and the nature of the owner of the IP address.
2. **Characterizing Authentication Schemes of Devices**: We analyzed the HTTP response codes of all devices in order to determine which smart-home devices authorized Shodan and which returned an authorization error. Successful HTTP responses indicated an unauthenticated or weakly authenticated device (e.g, just requiring a password).
3. **MQTT Payload data**: We also retrieved MQTT payload data from the device results that used MQTT. We specifically analyzed the MQTT connection codes as well as the MQTT topics devices displayed. As part of the MQTT Topics, we fetched the following information:
   (a) number of MQTT Topics observed by the device
   (b) the most commonly repeated topics among them
   (c) the maximum observed count of accepted topics by a broker

4. RESULTS

We discuss the results for the following measurement analyses: extent of accessibility of devices, i.e., how many devices have interfaces that are visible to the public Internet; from the actual devices extracted from the search results, what is the geographic footprint of these visible devices; and other characteristics of visible devices (HTTP and MQTT protocol characteristics like response and connection codes, topics etc.).

By studying how many devices have interfaces visible to the public Internet, we are studying the extent of accessibility of devices. How easily accessible a device is on a network could be considered a configuration issue rather than a vulnerability. This helped us measure how many of the devices are subject to careless configuration. We checked accessibility of each device in our original search result set either as having an accessible HTTP interface or reachable MQTT connection.

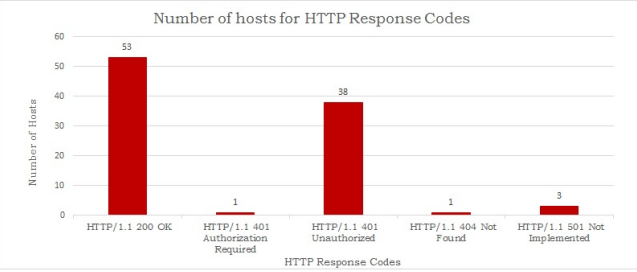

Figure 5: The distribution of HTTP response codes. HTTP 200 OK and HTTP 401 Authorization Required are the two most common responses. Both of these imply weak/no authentication mechanisms

Figure 5 shows that out of the 96 devices extracted having actual visible web interfaces, 53 returned the response 200 OK. This is an alarming amount of devices which were either allowing direct access to the device or had a simple login page for which credentials could potentially be easily brute-forced. 39 of the devices returned a 401 Authorization Required. This is an improvement over the login pages that we saw because even showing the login pages can reveal valuable information such as the device manufacturer and the model number which can be used to exploit implementation specific vulnerabilities. The authorization requirement is still a weakness as with sophisticated password crackers and devices using default configurations, it is possible to still obtain access to a device.

A simple query with the keyword “MQTT" revealed a host of connection vulnerabilities in devices and brokers. We analyzed the returned connection codes by the devices. The measurement revealed that out of all the results, 70 were accessed with connection code 0, as seen in Figure 6. This meant that when Shodan was performing network scans, it required no authorization to connect to them. The devices returned included both brokers and actual devices. The brokers which returned with connection code 0, have the potential to affect the behavior of all devices connected to that broker. Shodan also returned the list of topics that these brokers publish or stream content on. Many of these topics were seemingly standard topics supported by more than one broker device. A malicious device can advertise content to these commonly seen topics and can publish undesired content that could indirectly control multiple publishers. There is no limit to the number of topics a broker can support which can be exploited in many ways, e.g., flooding the broker with topics.

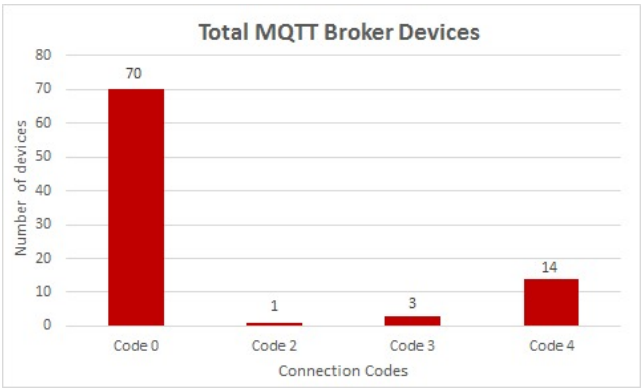

Figure 6: the distribution of MQTT connection codes. Code 0 was seen in most of the devices i.e. they did not require authorization to connect to them

Number of curated MQTT devices = 320
Topics observed = 455
Top 5 topics repeated: uC_RilduC_MainServer uC_MainServeruC_Rild ActiveMQ/Advisory/Consumer/Topic/# ActiveMQ/Advisory/MasterBroker devices/garage-temp-hum
Max observed topic count / broker = 138

Figure 7: Distribution of Topics in the returned MQTT devices

Table 3: Breakdown of Search Results by Continent

Continents	Prominent countries	Country Count	Device Count
Europe	Czech Republic, Germany, France	27	126
North America	United States	3	90
Asia	Taiwan, China, Korea	15	71
South America	Brazil	1	14
Australia	Australia	1	9
Africa	South Africa	1	3
Total		**48**	**313**

Table 3 displays the distribution of devices based on their geographic locations. Each continent had a few countries

which contributed to most of the devices found. Europe had the most number of visible devices. However, in North America, the United States contributed for about 90% of the visible devices. Interestingly, China, which is known for its widespread use of IoT devices contributes much less than expected to the total number of devices returned. This could be in part due to the fact that Internet usage is very restrictive in the country owing, as a result of which, a lot of connections may not be allowed.

Of the results extracted through the API, South America, Australia and Africa originated the least number of results. This could be because of the less widespread use of IoT devices in the countries of these continents as compared to Europe, United States, China and other countries where Smart-home IoT devices have gained popularity.

5. CHALLENGES & LIMITATIONS

The major stumbling blocks we faced were in terms of how to scope our searches on Shodan to return results for devices for which there were interesting vulnerabilities to inspect.

At the start of our project, we were exploring search results on Shodan with a broad array of search terms. These search terms corresponded to high-end wearable device brand names. This did not lead to very directed results or interesting results. To combat this, we expanded our scope from wearables to other small-scale IoT device types that have a higher potential to be directly connected to the Internet.

In terms of limitations, our study relied on data returned through the Shodan API. When making queries through the Shodan API, only a limited number (less than 100) of results are returned. Therefore, our measurements were only conducted on the limited set of results we retrieved per search term.

6. FUTURE WORK

Our current measurement study was performed on a small subset of results from Shodan available through the API. We would like to extrapolate our results and analysis to all the results from Shodan or a much larger dataset representative of the entire set of results.

Another area we would like to explore further is a risk analysis of vulnerable devices in the wild. We have identified a few types of devices whose interfaces are visible to the Internet. Given a type of device, we want to understand how many of these vulnerable devices are being used in the world. This information would give us an idea of the power of a possible attack exploiting a specific device's vulnerability. Additionally, we want to measure the bandwidth amounts that these vulnerable devices use. Understanding the bandwidth usage of devices can inform us about the attack potential of DDoS attacks using these devices.

Finally, as a broader application of our study, we would like to do a more comprehensive analysis of DDoS attacks using thingbots. Thingbots are embedded systems connected to the Internet that are compromised by a hacker to be used as bots, analogous to botnets compromising computers. Therefore, we feel it's important to study the attacks that have taken place using IoT thingbots as well as potential future attacks.

7. REFERENCES

[1] Blame the internet of things for destroying the internet today. https://motherboard.vice.com/en_us/article/blame-the-internet-of-things-for-destroying-the-inter

[2] Building smarter planet solutions with mqtt and ibm websphere mq telemetry. http://www.redbooks.ibm.com/redbooks/pdfs/sg248054.pdf.

[3] Open source tools available to assess risks to internet-facing ics.

[4] R. Bodenheim, J. Butts, S. Dunlap, and B. Mullins. Evaluation of the ability of the shodan search engine to identify internet-facing industrial control devices. *International Journal of Critical Infrastructure Protection*, 7(2):114 – 123, 2014.

[5] S. Hilton. Dyn analysis summary of friday october 21 attack.

[6] U. Hunkeler, H. L. Truong, and A. Stanford-Clark. Mqtt-sâĂŤa publish/subscribe protocol for wireless sensor networks. In *Communication systems software and middleware and workshops, 2008. comsware 2008. 3rd international conference on*, pages 791–798. IEEE, 2008.

[7] C. Lesjak, D. Hein, M. Hofmann, M. Maritsch, A. Aldrian, P. Priller, T. Ebner, T. Ruprechter, and G. Pregartner. Securing smart maintenance services: Hardware-security and tls for mqtt. In *Industrial Informatics (INDIN), 2015 IEEE 13th International Conference on*, pages 1243–1250. IEEE, 2015.

[8] E. P. Leverett. Quantitatively assessing and visualising industrial system attack surfaces. 2011.

[9] M. Long. I can show you the world: How shodan is used to exploit vulnerable scada systems.

[10] M. Paul M. Williams. Distinguishing internet-facing ics devices using plc programming information. 2014.

[11] M. Paul M. Williams. Internet of things ddos white paper. 2016.

[12] K. Simon. Vulnerability analysis using google and shodan. In *Cryptology and Network Security - 15th International Conference, CANS 2016, Milan, Italy, November 14-16, 2016, Proceedings*, pages 725–730, 2016.

[13] M. Singh, M. Rajan, V. Shivraj, and P. Balamuralidhar. Secure mqtt for internet of things (iot). In *Communication Systems and Network Technologies (CSNT), 2015 Fifth International Conference on*, pages 746–751. IEEE, 2015.