Analyzing Security and Privacy Concerns in MQTT Streams

Julia Odden
Northwestern University
juliaodden2022@u.northwestern.edu

Li Kang Tan
Northwestern University
litan2023@u.northwestern.edu

Adam Taranissi
Northwestern University
adamtaranissi2021@u.northwestern.edu

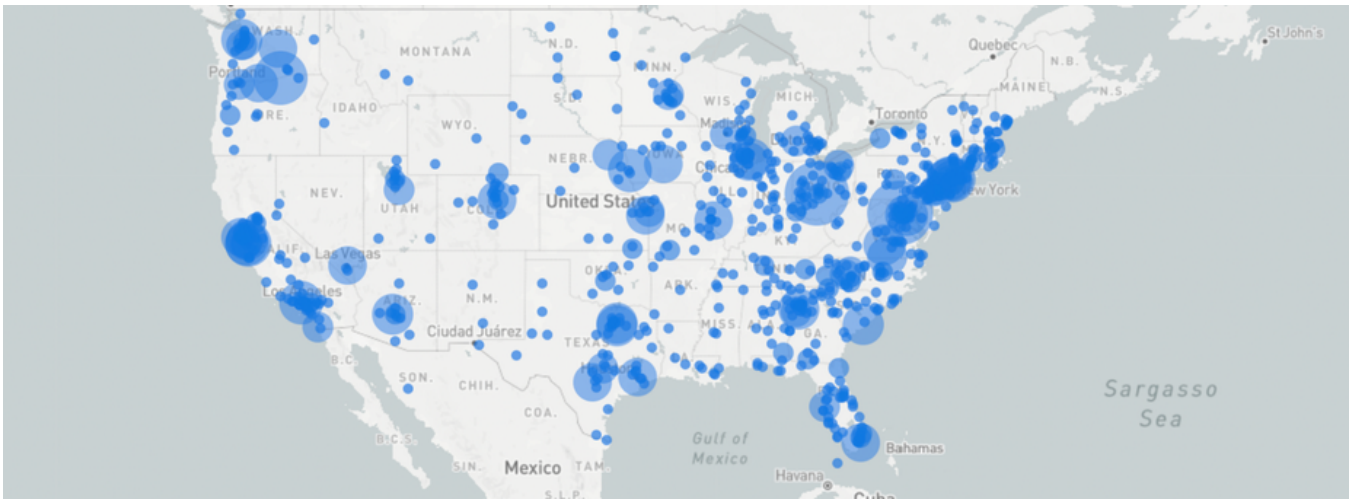

Figure 1: Distribution of US-based anonymously accessible MQTT devices.

ABSTRACT

Internet of Things (IoT) devices are becoming increasingly popular in both the commercial and private sectors for their convenience benefits. From smart door locks and baby monitors to hard hats and refrigerators, IoT devices are making the average person's life safer, more efficient, and smarter. There are over 35 billion IoT devices online worldwide [6]. In recent years, IoTs have received media attention as bugs and vulnerabilities have exposed users to eavesdropping and other attacks; these vulnerabilities pose an enormous risk to end users' security and privacy. This work analyzes and quantifies the scope of security and privacy leakage through the specific channel of anonymous connections to IoT devices.

1 INTRODUCTION

IoT devices are becoming ubiquitous, but are they safe? In our research, we seek to understand to what extent IoT devices leak sensitive information about their users into the public web. We examine two categories of data: private data, commonly referred to as personally identifiable information (PII), which includes information like emails or GPS coordinates; and security data, including access to alarm systems and home devices such as vacuums. Our contributions are as follows:

(1) A scraper tool to gather the IP addresses of IoT devices whose connections are anonymously available
(2) A tool to subscribe and publish to MQTT streams
(3) Quantitative and qualitative analysis of the data collected from our scraper and subscriber tools describing the extent of sensitive data leakage
(4) A geographical analysis of sensitive data leakage by IoT devices

2 BACKGROUND AND MOTIVATION

IoT devices do not send and receive messages directly to and from users' mobile devices. Instead, the IoT devices send their messages to a central broker that is shared among many devices. The broker then passes those messages along to the user's device. When the user wants to use their mobile phone to turn their lights off, they send a message to the broker, which then passes the message along to the IoT device(s) that subscribe(s) to the appropriate topic. E.g., if a user wanted to turn off their kitchen lights from their phone, they could send a message to the broker, which would then forward that message to any IoT devices that subscribe to the topic "myhouse/kitchenlights."

Many IoT devices and brokers use a protocol called MQTT to communicate. One of MQTT's many quirks is that devices using MQTT can be configured to accept connections from unknown devices, hereafter referred to as "anonymous connections." Unauthenticated users can subscribe to topics and publish messages on a broker as though they were legitimate users of the device. Not only does this allow anonymous users to control IoT devices that they do not own; it also permits anonymous users to snoop on the data passing through the broker, which often contains sensitive information such as location data or device schedules.

3 RELATED WORK

Unsurprisingly, a significant focus surrounding MQTT pertains to the security and privacy issues. For example, Antraper and Kotak have conducted research into security and privacy concerns of IoT devices [1]. They haven shown that it is possible to anonymously view messages on servers, publish (possibly malicious) messages to

brokers, and perform DoS attacks, thus compromising the confidentiality, integrity, and availability of an IoT service respectively. Our project builds on their research by performing similar analysis on a large scale. The examples used in the paper were isolated events, used to illustrate that such attacks were possible. In our research, we examine how prevalent these vulnerabilities truly are in the US.

Other research into the this area pertains to methods on securing and protecting MQTT systems. Colombo et al. propose an approach towards access control mechanisms for IoT ecosystems [2]. They prototyped a mechanism with Attribute-Based Access Control, and early experimentation has shown that the enforcement overhead is low, which is important given the nature of MQTT devices. However, users are often not at the forefront of such research. Our project thus indirectly quantifies the consumer uptake of security mechanisms in general, by examining the severity of security loopholes. As our results will show, the most securely designed system will not be effective if users simply do not use it. We hope that our project serves as a reminder that the weakest links in the system are often humans, and that technical approaches to securing IoT ecosystems must be backed up by proportionate effort in educating consumers and users so that the solutions discussed in research papers are implemented in real IoT ecosystems.

4 METHODOLOGY

The work of our project was divided into four distinct sections. First, we had to obtain a dataset of all of the IoT devices using MQTT that were open to anonymous connections. Next, we had to listen to the messages exchanged by the devices that allowed anonymous connection to amass our dataset of information flow. We then examined the messages, both automatically and manually, in order to conduct our analysis. Finally, we wrote a script that attempted to publish messages to open MQTT streams to determine whether or not an open stream could pose an immediate risk to the health and safety of the user.

4.1 Scraping for open MQTT connections

Our first step was to scan a representative subset of IP addresses for open MQTT connections. We leveraged the Shodan API [5] to accomplish this. MQTT uses port 1883, so we used a set of commands from the Shodan API that iterated through an arbitrary list of US-based IP addresses provided by Shodan and created a list of the IP addresses that returned a connection code of 0, indicating that our connection attempt had succeeded. Our requests are below.

```
> shodan download --limit 10000 /
results.json.gz port:1883 "MQTT connection /
code: 0" "country:US"
> shodan convert results.json.gz xlsx
```

4.2 Observing data flow

Our requests with the Shodan API yielded 6,720 IP addresses of devices using MQTT that permitted anonymous connections. We then wrote a Python program that connected to each IP address for five seconds and recorded the messages transmitted during that period. It is worth noting that some brokers retain messages for certain devices, so that if a new connection is made, they have information to send regarding that device, even if the device is not actively sending messages at the time the connection is made; other brokers are configured to only send messages that they have just received. As a result, not all of the 6,720 IP addresses we connected to returned data. Below is the script we used to generate this data automatically.

```
import pandas as pd
import paho.mqtt.client as mqtt
import json
import time
data = pd.read_excel("shodan_results.xlsx")
# callback when the client
# receives a CONNACK
def on_connect(client, userdata, flags, rc):
    print("Connected w/ result code "
                              +str(rc))
    client.subscribe("#")
# callback when a PUBLISH
# message is received
def on_message(client, userdata, msg):
    result = {}
    result["topic"] = msg.topic
    result["msg"] =
        str(msg.payload.decode("ASCII"))
    output = json.dumps(result)
    global fname
    f = open(fname, "a")
    f.write(output)
    f.close()
client = mqtt.Client()
client.on_connect = on_connect
client.on_message = on_message
fname = ""
for i in range(0, 50):
    fname = str(i) + ".txt"
    try:
        client.connect(data['IP'][i],
                              1883, 60)
    except:
        print("failed: ", data["IP"][i])
        continue
    starttime = time.time()
    client.loop_start()
    try:
        while True:
            if time.time() - starttime < 5:
                continue
            else:
                break
    except KeyboardInterrupt:
        print('KeyboardInterrupt')
    client.loop_stop()
```

4.3 Examining data

To examine the data generated by our Python script, we stored all of the individual data files in one folder and used Linux commands like grep, sort, and uniq to search for key terms and phrases such as "password," "username," "email," "lights," "GPS," "address," and others. This manual method was time-consuming but gave us fine-grained control over the information we recovered. We maintained a mapping of file index number to IP addresses and ran commands such as

```
> grep -l -r "[desired search term]" .
```

to return lists of files that contained that search term. We generated counts of key queries with a similar method:

```
> grep -l -r "[desired search term]" . /
    | grep -c "./"
```

4.4 Publishing messages

After examining the data we could receive as MQTT subscribers, we decided to attempt to publish our own messages. Notably, we only attempted to publish a message to a specific Amazon test broker; we did not push messages to actual users' MQTT devices. We recognize that this makes our report less representative of the true scope of this privacy issue, but the ethical issues of interfering with users' home lights systems or other personal devices outweighed our desire for thoroughly tested research conclusions. We suggest that a direction for further study would be to set up a testbed of IoT devices with anonymously accessible MQTT streams and attempt to modify or control them using the following script.

```
import pandas as pd
import paho.mqtt.client as mqtt
import time
data = pd.read_excel("shodan_results.xlsx")
# Callback for when the client receives
# a CONNACK response from the server.
def on_connect(client, userdata, flags, rc):
    print("Connected with result code "+
                                str(rc))
# Callback for when a PUBLISH message
# is received from the server.
def on_message(client, userdata, msg):
    print("\ntopic: ", msg.topic)
    print(str(msg.payload.decode("ASCII")))
    print("\nmessage qos: ",msg.qos)
    print("\nmessage retain flag: ",
                        msg.retain)
def on_publish(client,userdata,result):
    print("data published \n")
client = mqtt.Client()
client.on_connect = on_connect
client.on_message = on_message
client.on_publish = on_publish
client.connect(data['IP'][90], 1883, 60)
client.loop_start()
client.subscribe("[TOPIC REDACTED]")
result = client.publish("[TOPIC REDACTED]",
       "you might want to disable anonymous
       connections to your MQTT device")[0]
time.sleep(3)
client.loop_stop()
def on_disconnect(client, userdata, rc):
   print("client disconnected ok")
client.on_disconnect = on_disconnect
client.disconnect()
```

5 DATASETS

In our analysis, we examined two data-sets. The first was the list of IP addresses that our Shodan queries returned with a connection code of 0. We saved each response with a response code of 0 to an .xslx file referenced in the Python scripts above. The second data-set was the folder of .txt files containing five seconds' worth of connection data for each of the IP addresses in the .xlsx file. Those files are labeled by index in the .xslx file, so we could maintain a mapping between the data in a file and the IP address it corresponded to.

5.1 IP list

The data stored in the .xlsx file consisted of 17 columns: IP address, port number (always 1883), timestamp, a data payload (beginning with the connection code of 0, then following with the topic information and payload for that device), hostnames, organization, ISP, country, country ISO code, city, OS (usually empty), ASN, transport (always TCP), product (either MQTT or a version of Mosquitto), version (only of Mosquitto), web server, and website title. Generally, we were not interested in probing too deeply into any of the columns other than the IP addresses; the city labels were useful for our geographical analysis (and we used them to generate the figure at the top of this report), but this data-set primarily served as our master list of IP addresses we could connect to. We did notice that many of the devices were clustered around major corporate headquarters for Amazon.

5.2 Message files

We maintained a numbered message file for every IP address in the IP list. The name of the message file mapped directly to the row of the IP address in the IP list, so we could always identify the message file that corresponded to a given IP address, and vice versa. If a connection to an IP failed, we assume that device has been powered off or removed from the network, and its number is skipped.

The message files themselves contain simple JSON objects with two keys: "topic" and "message." Many message files have multiple JSON objects corresponding to multiple topics received within the span of one connection. A sample redacted version of one of these files is below.

```
{"topic": "link/linkUPStat", "msg":
"{\"interface_name\": \"eno4\", \"status\":
1, \"interface_probe_freq\": 5,
\"interface_alias\": \"\", \"interface_
```

```
display\": \"VSAT 1\", \"interface_type
\": \"vsat\", \"wan_updated_at\":
\"2022-03-09T02:41:03.981Z\", \"timeStamp\":
\"2022-03-09T02:41:03.981Z\", \"name\":
\"linkUPStat\",\"vesselId\": \"REDACTED\"}"}
```


6 RESULTS

The experiments we conducted on our data aimed to determine the scope of private and secure data leaked from anonymously accessible MQTT streams. We performed three experiments: a quantitative analysis of the message content, a back-tracing experiment to develop personal profiles of some of the device users, and a publication test in which we attempted to send a message to broker to update an IoT device via an anonymously available MQTT stream.

6.1 Analyzing message content

Our first experiment was intended judge the number of times sensitive information was transmitted through anonymously accessible MQTT streams. We divided our analysis into two types of data: private data and security information. Our methodology for this experiment is described in section 4.3. In addition to finding the number of instances of each search term, we also assigned each term a severity score in each category of privacy and security. A 1 indicates that the threat is not severe, while a 5 should be considered critical. We calculate each score by combining the risk factor of the search term (e.g., the appearance of "vacuum" is relatively low-risk, while "admin password" is very high risk) with the frequency at which it appeared (lower frequency indicates less risk). A sample of our data appears in table 1, below.

Term	Instances	Security	Privacy
light(s)	146	4	1
username	22	1	4
password	18	2	5
email	7	2	5
alarm	71	5	1
lock	103	5	1
longitude	64	5	5
door	107	5	1
fridge	7	2	2
camera	23	5	5
fire	14	5	1
sensor	255	4	4
baby	6	5	5
tesla	26	3	4
garage	60	5	2
vacuum	4	3	2
heater	21	3	3

Our results indicate that there is a significant amount of personal data leakage from open MQTT streams. We were able to see schedules for people's home lights, the battery percentage of their smart vacuums, their usernames and passwords to routers, their fire alarm information, and camera feed information. On its own, any of this information is compromising; when you consider that many of these terms were found together (e.g., "username" and "password" occurred together twice), these results are even more shocking.

We also sorted the anonymously accessible IoT devices by city, which was easy to extrapolate from the IP address of each device. The top ten cities are listed below in table 2.

City	Number
Ashburn	1052
Hilliard	951
Boardman	611
San Jose	279
North Bergen	257
Council Bluffs	249
Clifton	167
Santa Clara	158
Los Angeles	157
Hampden Sydney	142

Also high on the list were Dallas, New York City, Chicago, and Atlanta. The message analysis we performed demonstrates not just the scope of the data that we could observe, but also the nationwide spread of the issue. Our national findings suggest future work with an international data-set of IP addresses.

6.2 Back-tracing

After examining the message content, we attempted to develop user profiles for some of the IoT devices whose messages we could observe. Using some of the information in the files above, we created the following profiles. Sensitive information that we possess but should not share has been redacted.

Connection 1053 In this file, we could find a home address alongside an alarm schedule. At the time of our experiment, the alarm system was off.

Connection 81 This connection belonged to the K4 system of a yacht. K4 is a maritime internet management service [4] and the routers are on the IoT because they can be remotely controlled and configured through the user's cell phone. The MQTT stream contained the administrator login credentials (username and password) of the K4 system. It also contained the vessel ID, which we were able to back-trace to find the name of the yacht's owner. The stream also contained precise GPS coordinates, which told us exactly where the K4 device (and thus, the yacht) was at the moment of our experiment.

Connection 3294 This device stream contained username and password information for two network administration personnel. Since we were able to identify both people and one of their (we assume) commonly used passwords, we could have leveraged that information to log into their social media accounts.

Connection 2360 This connection belonged to a Guardhat, which is a hard hat designed for disaster [3]. The MQTT stream for this device contained body temperature measurements for the wearer, location data, and login information. It also contained a link to a camera feed, which we could access remotely. Using the camera feed and the location data, we were able to determine exactly where the construction worker was currently working; we even found the address of the house they were modifying on Zillow.

Connection 4506 This stream gave us access to somebody's smart home. We could see their lights and thermostat, and the stream contained the password to log into their home to make changes to their system.

Connection 1293 This connection gave us access to someone's front door lock, including its current state of open or closed. We could also identify the manufacturer and serial number of the lock device and the state of the device's battery.

In addition to each of these connections, we were able to access two live camera feeds. We also had the ability to log into several databases with administrator credentials. For ethical reasons, we did not attempt this.

6.3 Publishing messages

It is one thing to be able to view the status of and information about a user and their smart devices. It is an entirely different issue if we are able to modify it via an anonymous connection. The script in section 4.4 attempts to connect to a broker and use it to publish a message to an IoT device as though we were authorized users. We successfully tested the script on a single Amazon test instance of a device with an open MQTT stream. In that script, we modified the content of one arbitrary topic on the device to hold the string "You might want to disable anonymous connections to your MQTT device." We can tell that our message was published successfully because it was returned to us when we set ourselves up to also behave as a subscriber.

We did not conduct further tests with this script because we did not want to risk sending a message that would damage someone's IoT device or potentially jeopardize their safety (in the case of a door lock or GuardHat).

7 IMPLICATIONS

The first and most obvious implication of our findings comes from the types of data we could see, and the information we could extrapolate from it, in section 6.2. The amount of information we were able to determine about the IoT device users indicates that using the MQTT protocol with anonymous connections enabled is not acceptable. We believe that there is not nearly enough public awareness of the possible security risks of using an IoT device, particularly a security-related device such as a smart lock or a smart baby monitor. Too many people are willing to accept the convenience-privacy trade-off, possibly because they are not even aware that an exchange is taking place. Who would design a smart lock that has anonymous remote connection enabled? It simply does not make sense, and thus, the average user likely does not consider the scenario, especially not when overcome with the excitement over a new and convenient home device. Additionally, the availability of anonymously connectable MQTT streams is often due to user setup error, indicating that many of these devices do not provide robust instructions on how to set them up to be secure.

The second implication of this work comes from combining the information on the types of available IoT devices in sections 6.1 and 6.2 with the connection and publishing results in section 6.3. Knowing that we can access people's smart locks, and that we can arbitrarily write messages to IoT devices via their brokers, there is nothing but ethics preventing us from opening those door locks from our desks in Chicago. We could spontaneously change the login information for the K4 device described in section 6.2 and cause the user's yacht to lose Internet connection while at sea. We could disable fire detectors, smoke detectors, alarm systems, or baby monitors; open garages; crank up heaters or cool them down; heat refrigerators to ruin groceries; or spontaneously turn lights and vacuums on and off like poltergeists. Anonymous communication permits anonymous tampering. Hackers could use these open MQTT streams to do irreparable harm to IoT users without ever needing a password.

8 LIMITATIONS

Our work, and particularly our testing, was severely limited by ethics and by lack of funding. First and foremost, we would have liked to perform international scans with the Shodan API and to not limit ourselves to 10,000 data points; however, Shodan charges by IP address scanned, and we did not have enough academic credits to perform wider scans. As it stands, we were forced to limit our dataset to at most 10,000 points within the United States.

Because we were dealing with users' personal devices, we also were unable to thoroughly test our message publication script. We did not want to, for example, unlock someone's home to verify that our script worked on residential devices. We have not tested the script on any devices aside from a test device named and designated by Amazon. We also have not leveraged any of the personal data we discovered as part of our experiment in section 6.3 to determine if it is valid. For example, we made no attempt to modify settings on the K4 device despite having administrator credentials.

9 FUTURE WORK

We suggest four directions for future work on this topic.

(1) International data. We suggest that future research teams either employ an alternative to Shodan.io that does not charge by connection or pay Shodan.io for enough connections to perform a more robust international search. A larger dataset would enable researchers to generate better statistics, particularly on types of connections like the ones indicated in the table in section 6.1. It would also be interesting to generate statistics on what percentage of IP addresses touched by the scraper enabled versus blocked anonymous connections.
(2) IoT device testbed. Future researchers should consider purchasing their own IoT devices and configuring them to accept anonymous MQTT connections. They should then test the script given in section 4.4 on those devices to determine whether or not it is generally effective. Having a testbed of private IoT devices eliminates the ethical considerations of using an anonymous connection and a script to alter their settings.
(3) Quantitative metrics for private and secure data leakage. The security and privacy data leakage metrics in the third and fourth columns of the table in section 6.1 provide a general picture of the severity of the data leakage on each topic, but are not particularly empirical. We suggest that in the future, researchers devise or adapt a method to more accurately quantify data leakage in each category. It would

also be interesting to measure data leakage based on type of device; e.g., determining which IoT devices and manufacturers have the highest rates of privacy and security violations. This research could easily translate to a buying guide for IoT devices.

(4) Raising awareness of the issue. Clearly, the presence of open MQTT streams that can be used to connect to and modify IoT devices is a serious issue that lacks significant public awareness. Future researchers and activists could campaign for better security and privacy awareness among the consumer public and higher levels of accountability from IoT manufacturers.

10 CONCLUSION

Smart homes and smart devices are extremely convenient, but perhaps it is time for people to swing back in the direction of privacy over convenience. There is a saying in computer science that the more you know about security, the more you know it is a myth. Another adage jokingly states that a cybersecurity professional keeps only two pieces of technology in their home: a printer for work and a loaded gun beside the printer in case it makes a funny noise. As silly and paranoid as these may sound, they are grounded in the unfortunate reality that introducing smart technology into our lives leads to privacy and security violations. There is a clear tradeoff between privacy and convenience, and due in part to a lack of knowledge, many consumers are erring on the side of convenience. The authors encourage readers to research their devices before buying, and if no information on the privacy or security of the device is readily available or comprehensible, we invite you to imagine us at our computers viewing your data before choosing to install the smart lock or set up the smart refrigerator. In the privacy-convenience tradeoff, it is all too likely that the short-term convenience of a smart device will be outweighed in the long run by the privacy or security violation it permits.

ACKNOWLEDGMENTS

We thank Dr. Sruti Bhagavatula for her mentorship during this project.

REFERENCES

[1] Joseph Jose Anthraper and Jaidip Kotak. 2019. *Security, Privacy and Forensic Concern of MQTT Protocol.* Retrieved March 15, 2022 from https://papers.ssrn.com/sol3/papers.cfm?abstract_id=3355193

[2] Pietro Colombo and Elena Ferrari. 2018. *Access Control Enforcement within MQTT-based Internet of Things Ecosystems.* Retrieved March 15, 2022 from https://dl.acm.org/doi/pdf/10.1145/3205977.3205986

[3] Guardhat. 2022. *Guardhat.* Retrieved March 8, 2022 from https://www.guardhat.com/

[4] K4 Mobility. 2022. *K4 Mobility Solutions.* Retrieved March 8, 2022 from https://www.k4mobility.com/

[5] Shodan.io. 2022. *Shodan API.* Retrieved February 15, 2022 from https://developer.shodan.io/

[6] Jack Steward. 2022. *The Ultimate List of Internet of Things Statistics for 2022.* Retrieved March 10, 2022 from https://findstack.com/internet-of-things-statistics/