IOT Research & Innovation Lab (IRIL)
Sultan Qaboos IT Chair, KICS, UET Lahore

IoT Advanced Data Generator

V 0.0.1

01/01/2019

About

Internet of Things (IoT) is new trend. Researchers need IoT data for their analysis. Most of IoT data generators publically available are web based/cloud based and providing limited functionality. Functionality extension facility is difficult in existing IoT data generators because of their codes are not public.

"**IoT Advanced Data Generator**" allows users to create complete IoT use case on single machine with traffic capturing ability on same machine. **Compromised IoT device** is new concept provided by IoT Advanced Data Generator, which is missing in existing IoT data generators. Also **New protocols/functionality** can easily be added in IoT Advanced Data Generator. User can create use cases and add devices in those uses cases with different IP. Complete network traffic can be logged using wire shark on same machine.

IoT Data Generator	COAP Support	MQTT Support	Separate IP Address	Static / Dynamic Data generation	Periodic / Random Data generation	Attacking Entity	Platform	Open Source
Simple IoT Simulator	✓	✓	✓	✓	✓	✘	Linux	✘
Node-RED	✘	✓	✓	NA	✓	✘	Multiple	✓
Things Board	✓	✓	NA	NA	✓	✘	Multiple	✓
IOTIFY	✓	✓	✓	NA	✓	✘	online	NA
IBM Bluemix	✘	✓	NA	NA	✓	✘	Linux, Online	Free, Paid
Microsoft Azure IoT	✘	✘	✓	NA	✓	✘	online	NA
NetSim	✘	✘	NA	NA	✓	✓	Windows	✘
BevyWise IoT	✘	✓	NA	NA	✓	✘	Multiple	NA
IoT Advanced Data Generator	✓	✓	✓	✓	✓	✓	Linux	✓

Table 1: Data Generator Comparison

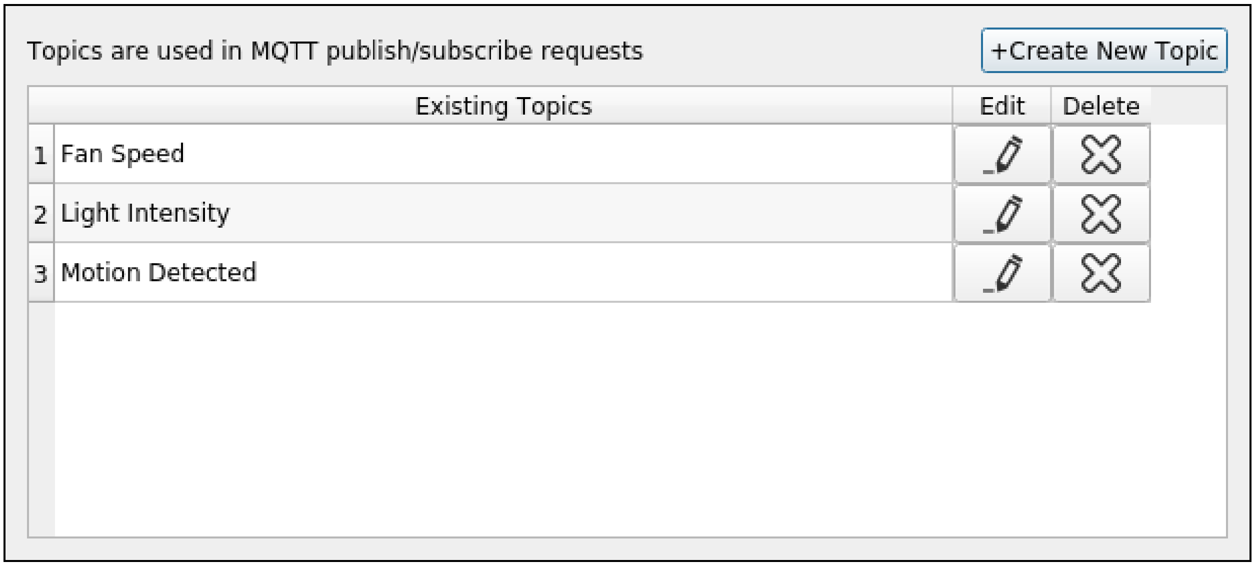
Table of Contents

1. Setup

Setup section allows users to create templates of information. These templates are used in application avoiding typing information again and again.

1.1 Topic

Topics are used in MQTT protocol implementation. Devices using MQTT protocol can be publisher/scriber of topics. Click on main menu **Setup->Topic.** Already Created topics will appear on screen below. Topic edit/delete functionality is provided on this screen.

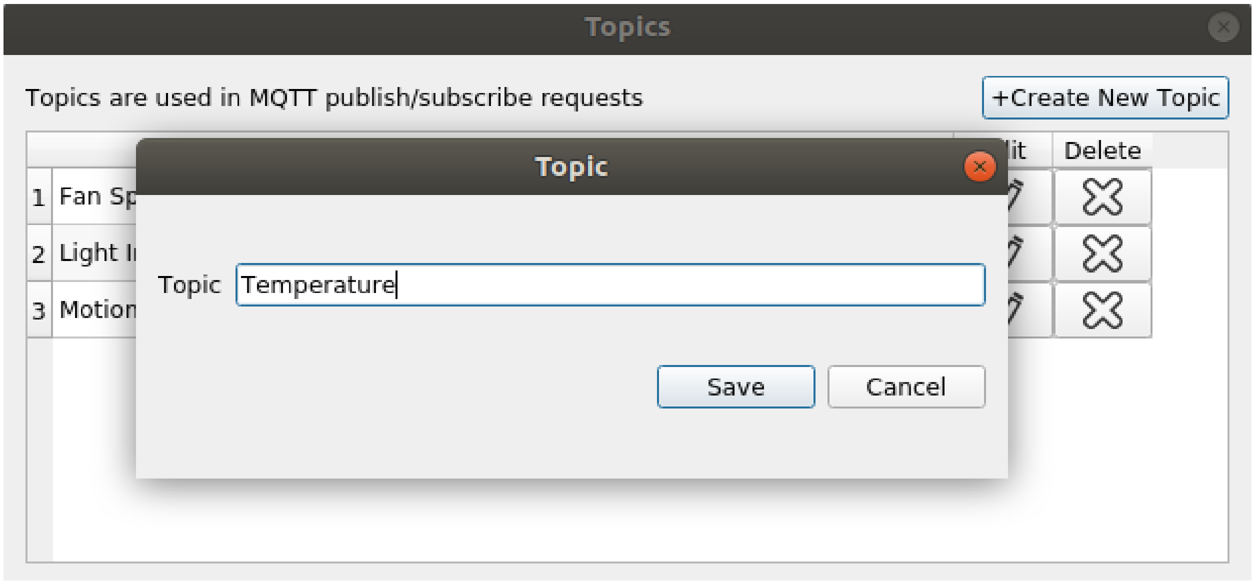


Topics are used in MQTT publish/subscribe requests

	Existing Topics	Edit	Delete
1	Fan Speed	✎	✕
2	Light Intensity	✎	✕
3	Motion Detected	✎	✕

Fig 1: Topic view/edit/delete

Click on "**+Create New Topic**" for adding new topic.



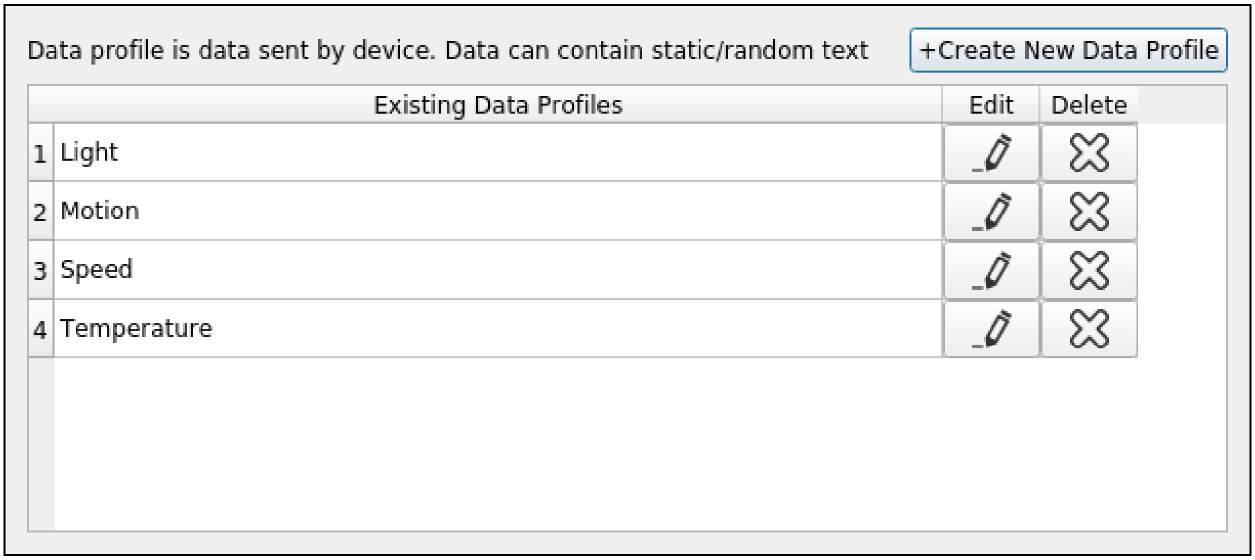
Topics are used in MQTT publish/subscribe requests

Topic: Temperature

Fig 2: New topic

1.2 Data Profile

IoT devices send data that can be static/random. Data Profile screen allow users to create data generation capability that real time IoT devices actually produce. Click on main menu **Setup->Data Profile**. Already Created data profiles will appear on screen below. Data Profile edit/delete functionality is provided on this screen.



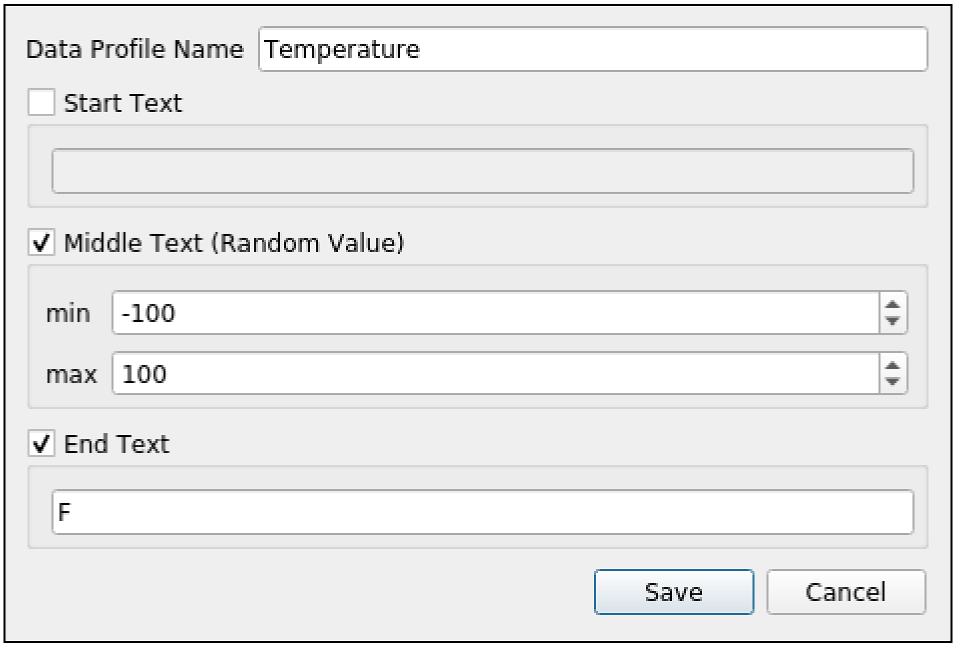
Data profile is data sent by device. Data can contain static/random text

+Create New Data Profile

	Existing Data Profiles	Edit	Delete
1	Light	✎	✖
2	Motion	✎	✖
3	Speed	✎	✖
4	Temperature	✎	✖

Fig 3: Data profile view/edit/delete

Click on "**+Create New Data Profile**" for adding new data profile.



Data Profile Name: Temperature

□ Start Text

☑ Middle Text (Random Value)

min: -100

max: 100

☑ End Text

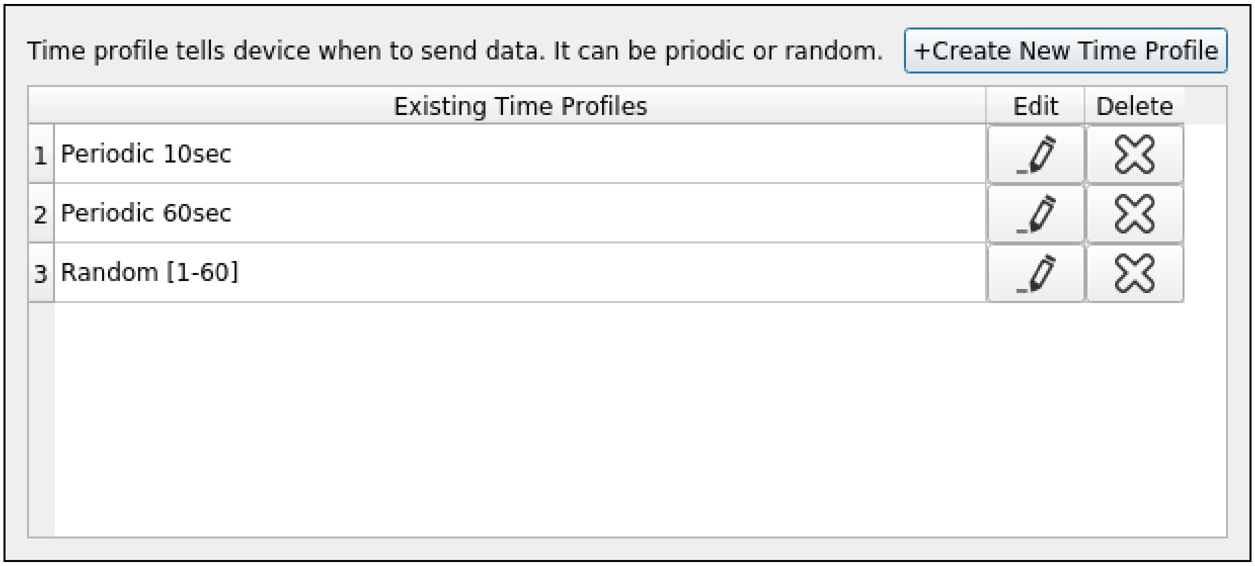
F

Save | Cancel

Fig 4: New data profile

1.3 Time Profile

IoT devices send information periodic or based on some events. User can create periodic/random time profiles. Click on main menu **Setup->Time Profile**. Already created time profiles will appear on screen below. Time Profile edit/delete functionality is provided on this screen.

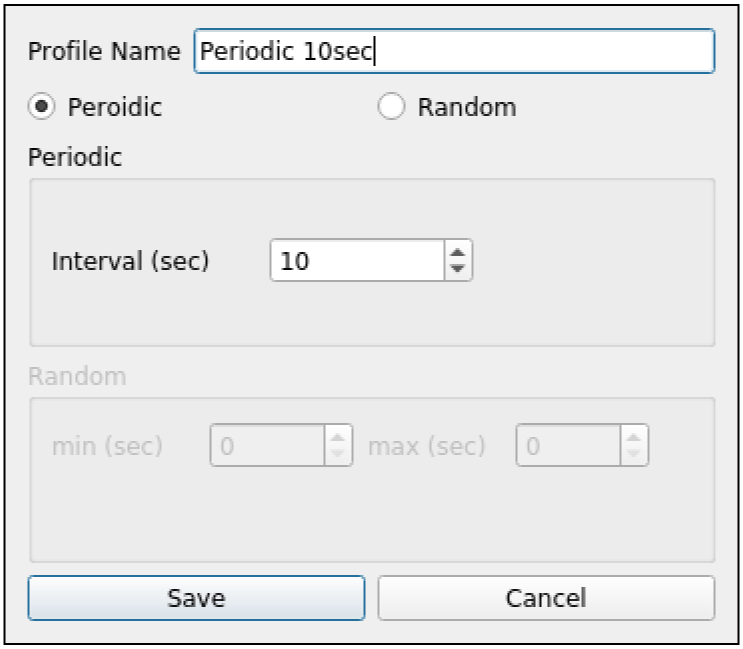


Time profile tells device when to send data. It can be priodic or random.

	Existing Time Profiles	Edit	Delete
1	Periodic 10sec	✎	✕
2	Periodic 60sec	✎	✕
3	Random [1-60]	✎	✕

Fig 5: Time profile view/edit/delete

Click on "**+Create New Time Profile**" for adding new data profile



Profile Name: Periodic 10sec

● Peroidic ○ Random

Periodic

Interval (sec): 10

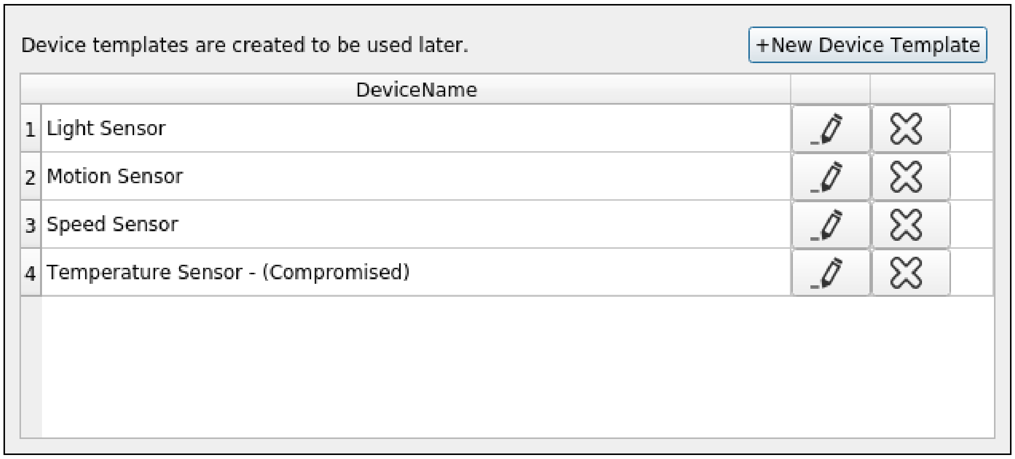
Random

min (sec): 0 max (sec): 0

Fig 6: New time profile

1.4 Device Template

IoT devices configuration like protocol setting, IP setting etc is done in device template area. Click on main menu **Setup->Device Template**. Already created device templates will appear on screen below. Device Template edit/delete functionality is provided on this screen.



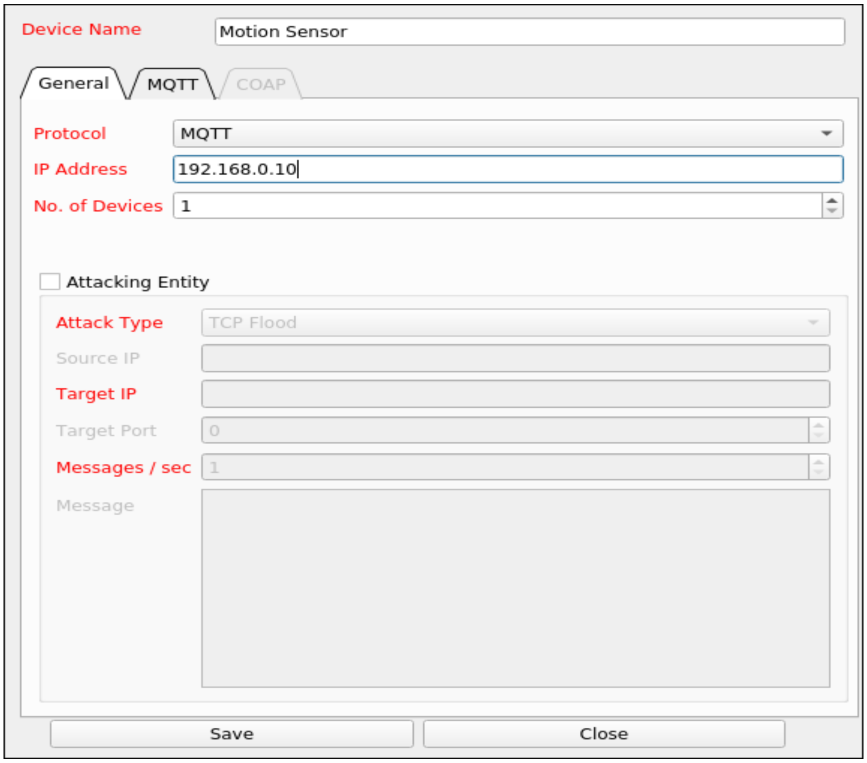
Device templates are created to be used later.

	DeviceName
1	Light Sensor
2	Motion Sensor
3	Speed Sensor
4	Temperature Sensor - (Compromised)

Fig 7: Device template view/edit/delete

1.4.1 Creating MQTT device

Click on **"+New Device Template**" for creating new device template.



Device Name: Motion Sensor

Protocol: MQTT

IP Address: 192.168.0.10

No. of Devices: 1

☐ Attacking Entity

Attack Type: TCP Flood

Source IP:

Target IP:

Target Port: 0

Messages / sec: 1

Message:

Fig 8: Creating MQTT device

Device Name: Name of device template

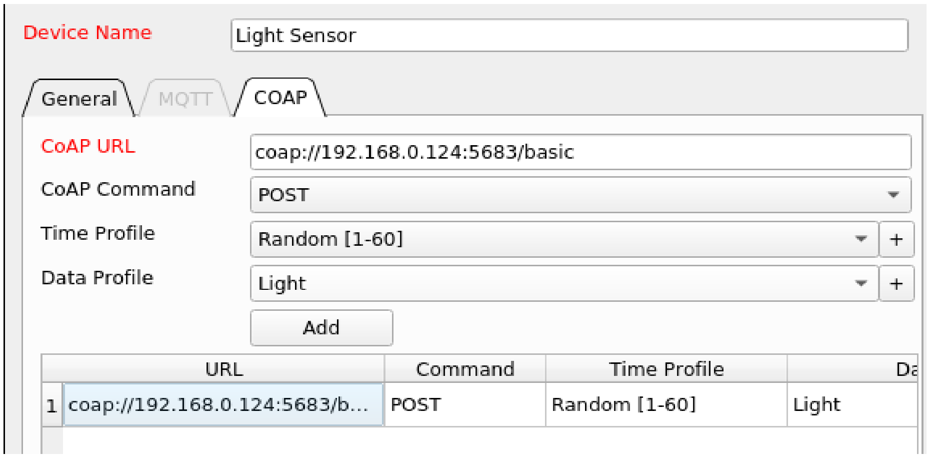
Protocol: MQTT

IP Address: Provide valid IP address, it will be assigned to device.

No. of Devices: Application will automatically create enter number of devices with incremented IP Address.

1.4.2 Creating COAP device

Similar to MQTT device creation, COAP device can be created by selecting Coap protocol from general tab.



Device Name: Light Sensor

CoAP URL: coap://192.168.0.124:5683/basic

CoAP Command: POST

Time Profile: Random [1-60]

Data Profile: Light

	URL	Command	Time Profile	Da
1	coap://192.168.0.124:5683/b...	POST	Random [1-60]	Light

Fig 8: Creating COAP device

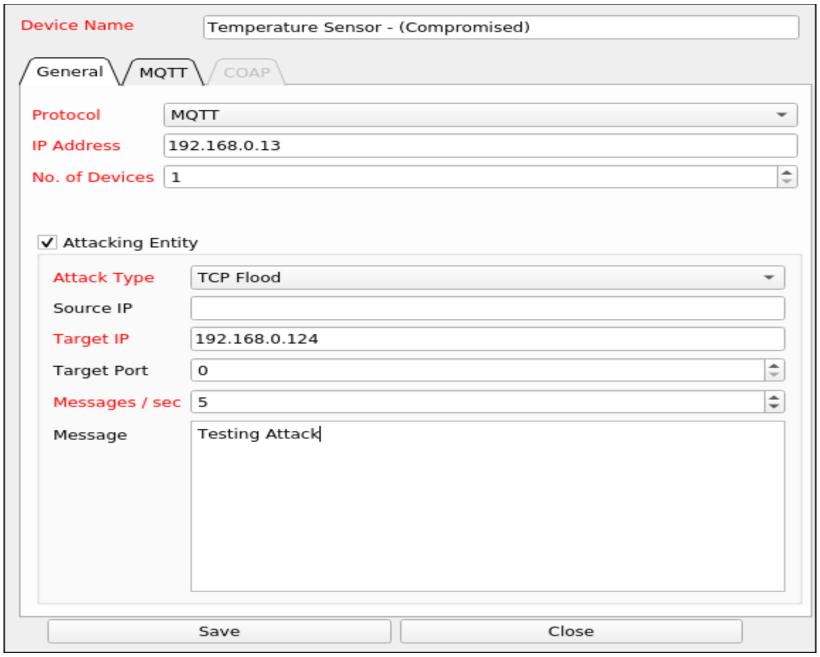
Coap URL: url of COAP server which can process coap requests.

Coap Command: it can be GET, PUT, POST, DELETE

**Enter the required fields and click on Add button for saving COAP device.

1.4.3 Compromised Device

Compromised device is new concept that is missing in available IoT data generators. Using compromised device application can generate normal device traffic along with attack traffic. Such traffic can be analyzed to capture compromised device.



Device Name: Temperature Sensor - (Compromised)

Protocol: MQTT

IP Address: 192.168.0.13

No. of Devices: 1

☑ Attacking Entity

Attack Type: TCP Flood

Source IP:

Target IP: 192.168.0.124

Target Port: 0

Messages / sec: 5

Message: Testing Attack

Fig 9: Creating Compromised device

Attack Type: TCP Flood, UDP Flood

Source IP: If source IP provided then packet sent with spoof IP, otherwise IP address of device is used.

Messages/sec: After every second number of messages sent to target IP

Target IP: Device towards which attack generated.

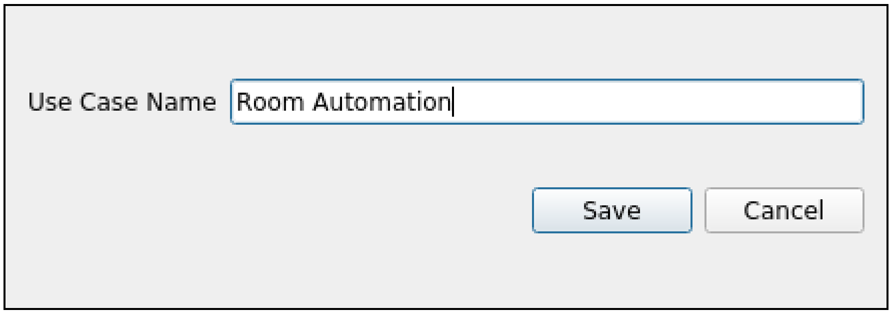
Message: Payload of attack packet.

2. Use Case

Users can create use case like home automation; safe city etc and captured traffic can help in doing analysis and other research tasks for IoT traffic.

2.1 Creating New Use Case

For creating new use case click on main menu “**Setup->Use Case**”. Screen showing existing use cases will appear. Click on “**+New Use Case”** for creating new use case.

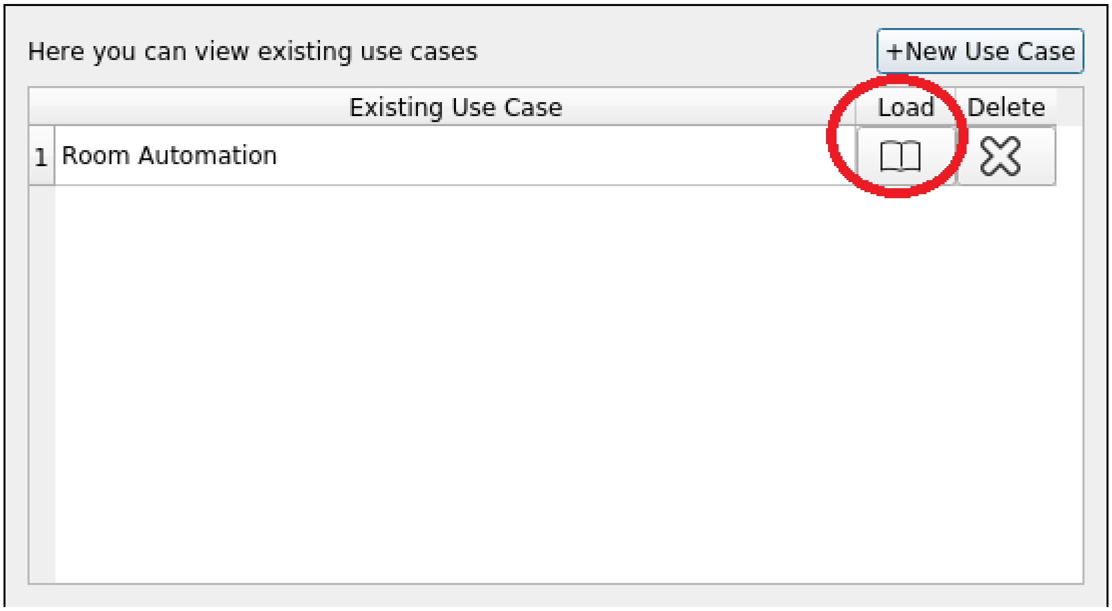


Use Case Name: Room Automation

Fig 10: New use case

2.2 Loading Use Case

Use case need to load to add devices in it. From use case screen click on load button shown below.



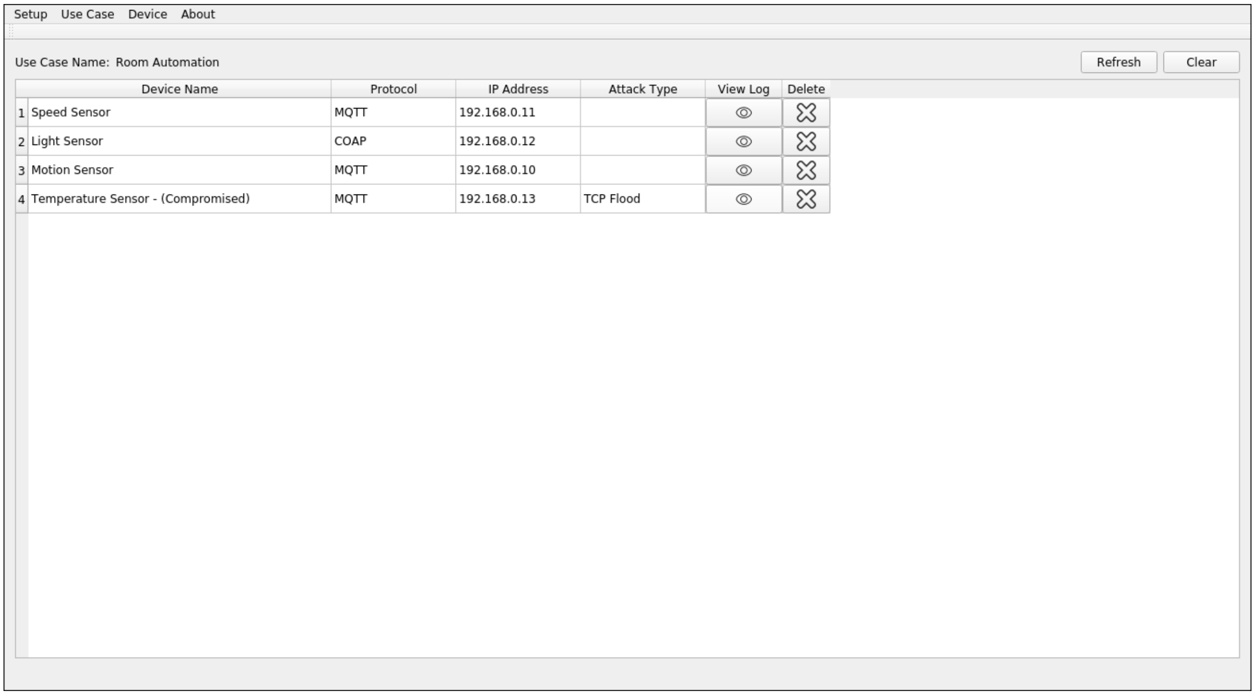
Here you can view existing use cases

	Existing Use Case	Load	Delete
1	Room Automation	🕮	✖

Fig 11: View/Load/Delete Usecase

2.3 Adding Devices in Use Case

Selected use case will be loaded on main screen shown below. You can add devices in use case using main menu “**Device->New Device**”. Existing device templates will also appear on same new device screen.



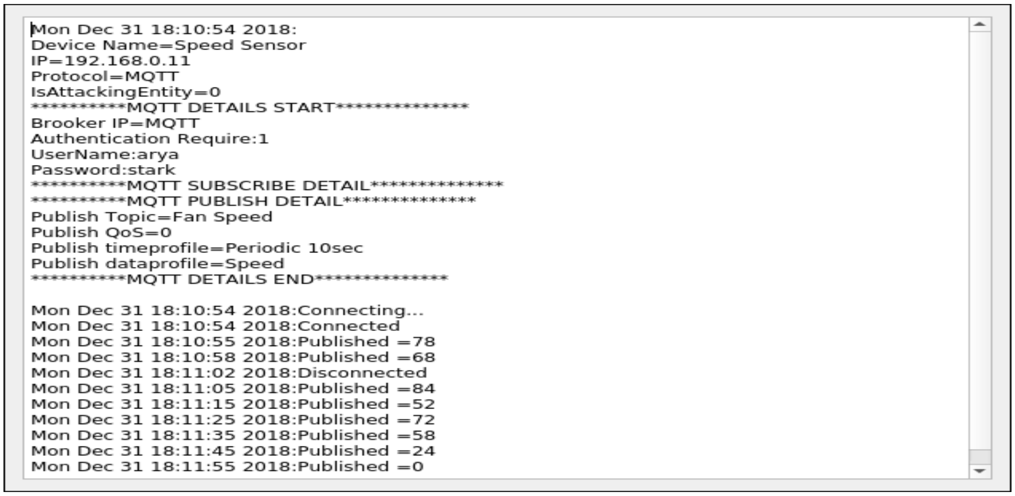
Use Case Name: Room Automation

	Device Name	Protocol	IP Address	Attack Type	View Log	Delete
1	Speed Sensor	MQTT	192.168.0.11		👁	☒
2	Light Sensor	COAP	192.168.0.12		👁	☒
3	Motion Sensor	MQTT	192.168.0.10		👁	☒
4	Temperature Sensor - (Compromised)	MQTT	192.168.0.13	TCP Flood	👁	☒

Fig 12: Home screen

2.4 View Log

For each device in use case clicking on “View Log” button shown in above screen, user can view log.



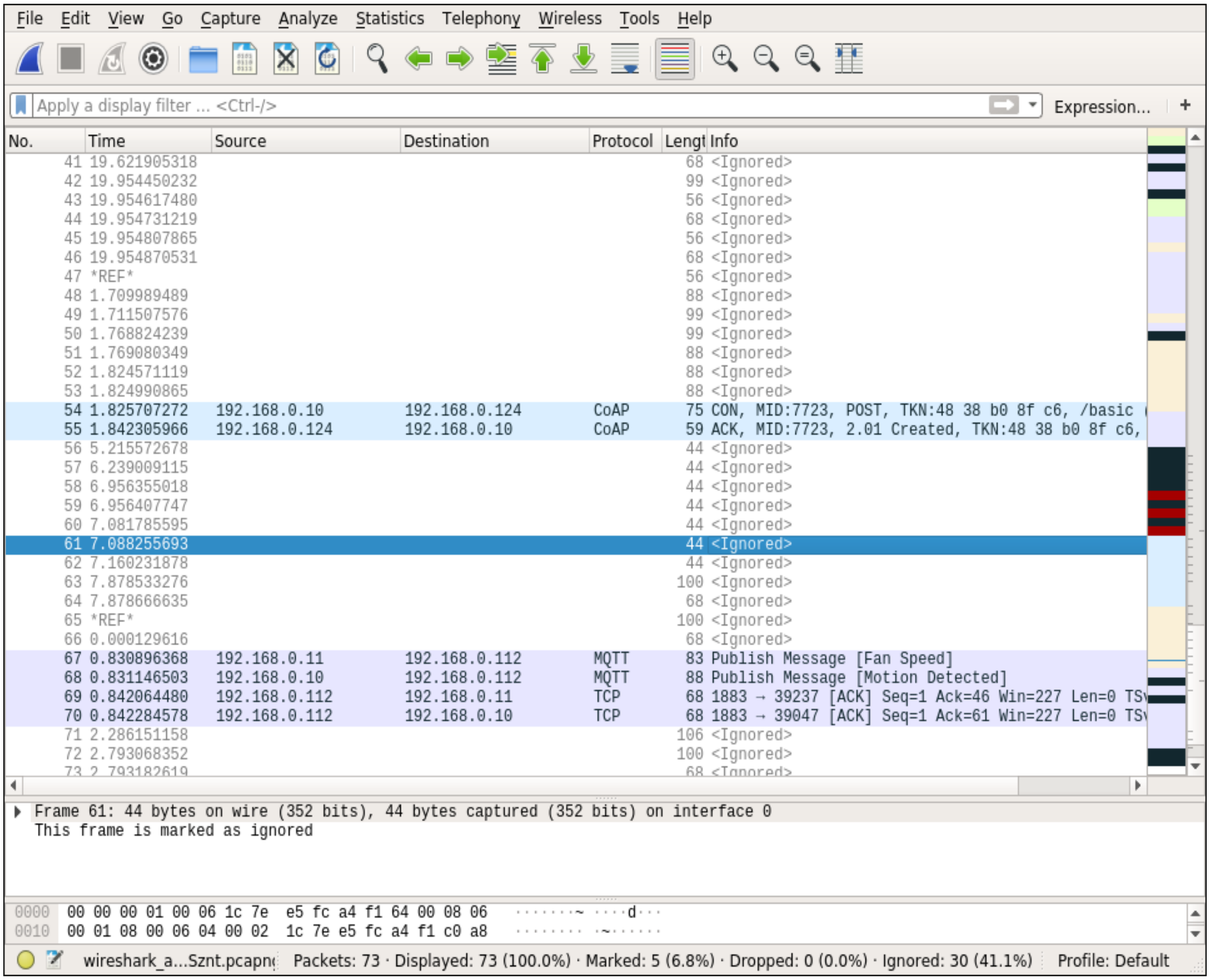
```
Mon Dec 31 18:10:54 2018:
Device Name=Speed Sensor
IP=192.168.0.11
Protocol=MQTT
IsAttackingEntity=0
***********MQTT DETAILS START***************
Brooker IP=MQTT
Authentication Require:1
UserName:arya
Password:stark
***********MQTT SUBSCRIBE DETAIL***************
***********MQTT PUBLISH DETAIL***************
Publish Topic=Fan Speed
Publish QoS=0
Publish timeprofile=Periodic 10sec
Publish dataprofile=Speed
***********MQTT DETAILS END***************

Mon Dec 31 18:10:54 2018:Connecting...
Mon Dec 31 18:10:54 2018:Connected
Mon Dec 31 18:10:55 2018:Published =78
Mon Dec 31 18:10:58 2018:Published =68
Mon Dec 31 18:11:02 2018:Disconnected
Mon Dec 31 18:11:05 2018:Published =84
Mon Dec 31 18:11:15 2018:Published =52
Mon Dec 31 18:11:25 2018:Published =72
Mon Dec 31 18:11:35 2018:Published =58
Mon Dec 31 18:11:45 2018:Published =24
Mon Dec 31 18:11:55 2018:Published =0
```

Fig 12: Device log from “IoT Advanced Data Generator”

3. Network Traffic Logging

User case network traffic can be logged using wire shark. Example of logged traffic is shown below.



```
No.      Time          Source          Destination      Protocol  Lengt Info
      41 19.621905318                                              68 <Ignored>
      42 19.954450232                                              99 <Ignored>
      43 19.954617480                                              56 <Ignored>
      44 19.954731219                                              68 <Ignored>
      45 19.954807865                                              56 <Ignored>
      46 19.954870531                                              68 <Ignored>
      47 *REF*                                                     56 <Ignored>
      48 1.709989489                                               88 <Ignored>
      49 1.711507576                                               99 <Ignored>
      50 1.768824239                                               99 <Ignored>
      51 1.769080349                                               88 <Ignored>
      52 1.824571119                                               88 <Ignored>
      53 1.824990865                                               88 <Ignored>
      54 1.825707272   192.168.0.10    192.168.0.124    CoAP       75 CON, MID:7723, POST, TKN:48 38 b0 8f c6, /basic
      55 1.842305966   192.168.0.124   192.168.0.10     CoAP       59 ACK, MID:7723, 2.01 Created, TKN:48 38 b0 8f c6,
      56 5.215572678                                               44 <Ignored>
      57 6.239009115                                               44 <Ignored>
      58 6.956355018                                               44 <Ignored>
      59 6.956407747                                               44 <Ignored>
      60 7.081785595                                               44 <Ignored>
      61 7.088255693                                               44 <Ignored>
      62 7.160231878                                               44 <Ignored>
      63 7.878533276                                              100 <Ignored>
      64 7.878666635                                               68 <Ignored>
      65 *REF*                                                    100 <Ignored>
      66 0.000129616                                               68 <Ignored>
      67 0.830896368   192.168.0.11    192.168.0.112    MQTT       83 Publish Message [Fan Speed]
      68 0.831146503   192.168.0.10    192.168.0.112    MQTT       88 Publish Message [Motion Detected]
      69 0.842064480   192.168.0.112   192.168.0.11     TCP        68 1883 → 39237 [ACK] Seq=1 Ack=46 Win=227 Len=0 TS
      70 0.842284578   192.168.0.112   192.168.0.10     TCP        68 1883 → 39047 [ACK] Seq=1 Ack=61 Win=227 Len=0 TS
      71 2.286151158                                              106 <Ignored>
      72 2.793068352                                              100 <Ignored>
      73 2.793182619                                               68 <Ignored>
```

```
▸ Frame 61: 44 bytes on wire (352 bits), 44 bytes captured (352 bits) on interface 0
  This frame is marked as ignored
```

```
0000  00 00 00 01 00 06 1c 7e  e5 fc a4 f1 64 00 08 06   ·······~ ····d···
0010  00 01 08 00 06 04 00 02  1c 7e e5 fc a4 f1 c0 a8   ········ ·~······
```

wireshark_a...Sznt.pcapn(Packets: 73 · Displayed: 73 (100.0%) · Marked: 5 (6.8%) · Dropped: 0 (0.0%) · Ignored: 30 (41.1%) Profile: Default

Fig 13: Network traffic logging using wireshark