System Security (201700086)
IoT #2 - active attacks against the smart plug

Æde Symen Hoekstra, Luigi Coniglio (Group 24)

May 21, 2018

1 Setup

For this assignment we were asked to perform a MITM attack against an IoT device. The target of the attack was a smart plug using an encrypted connection.

At startup the device connects to the cloud server using an untrusted HTTPS connection, for this reason we were able to intercept and decrypt the communication by providing a crafted certificate to the device.

Together with this document we also provided a capture containing all the steps covered in section 2.

2 Capture

In this section we show the transmitted data for when the switch connects to the cloud, when it is switched on and when it is switched off. As we can see the device and the server use JSON-encoded data.

2.1 Device Start-up

When the switch is powered on and first connects to the cloud it will initiate a WebSocket opening handshake.
Request:

```
GET /api/ws HTTP/1.1
Host: iotgo.iteadstudio.com
Connection: upgrade
Upgrade: websocket
Sec-WebSocket-Key: ITEADTmobiM0x1DabcEsnw==
Sec-WebSocket-Version: 13
```

Reply:

```
HTTP/1.1 101 Switching Protocols
upgrade: websocket
```

```
connection: upgrade
sec-websocket-accept: q1/L5gx6qdQ7y3UWgO/Tah/zXBA=
```

After the WebSocket-connection has been initiated the device will register itself on the cloud transmitting this message:

```
{
   "userAgent":"device",
   "apikey":"7442c80d-838d-4587-b191-43516e9d5f96",
   "deviceid":"1000248f2d",
   "action":"register",
   "version":2,
   "romVersion":"1.5.5",
   "model":"PSA-B01-GL",
   "ts":744
}
```

The cloud server will then reply providing some configuration:

```
{
   "error":0,
   "deviceid":"1000248f2d",
   "apikey":"82fcc3f7-072c-49ab-860f-f8a20fb9f754",
   "config":{
      "devConfig":{
         "storeAppsecret":"",
         "bucketName":"",
         "lengthOfVideo":0,
         "deleteAfterDays":0,
         "persistentPipeline":"",
         "storeAppid":"",
         "uploadLimit":0,
         "statusReportUrl":"",
         "storetype":0,
         "callbackHost":"",
         "persistentNotifyUrl":"",
         "callbackUrl":"",
         "persistentOps":"",
         "captureNumber":0,
         "callbackBody":""
      },
      "hb":1,
      "hbInterval":145
   }
}
```

2.2 Device Powered on

When the application issues a power-on command, the cloud will instruct the device to switch-on as follows:

```
{
   "action":"update",
   "deviceid":"1000248f2d",
   "apikey":"82fcc3f7-072c-49ab-860f-f8a20fb9f754",
   "userAgent":"app",
   "sequence":"1526577496475",
   "ts":0,
   "params":{
      "switch":"on"
   },
   "from":"app"
}
```

The device will then (in case of success) reply:

```
{
"error":0,
"userAgent":"device",
"apikey":"82fcc3f7-072c-49ab-860f-f8a20fb9f754",
"deviceid":"1000248f2d",
"sequence":"1526577496475"
}
```

2.3 Device Powered off

When the application issues a power-off command, the cloud will instruct the device to switch-off as follows:

```
{
   "action":"update",
   "deviceid":"1000248f2d",
   "apikey":"82fcc3f7-072c-49ab-860f-f8a20fb9f754",
   "userAgent":"app",
   "sequence":"1526577542585",
   "ts":0,
   "params":{
      "switch":"off"
   },
   "from":"app"
}
```

The device will then (in case of success) reply:

```
{
   "error":0,
   "userAgent":"device",
   "apikey":"82fcc3f7-072c-49ab-860f-f8a20fb9f754",
   "deviceid":"1000248f2d",
```

```
6       "sequence":"1526577542585"
7   }
```