Table of Contents

1. Hacking the Silvercrest (Lidl) Smart Home Gateway 2
1.1. 1. Connecting the device 2
1.2. 2. Getting root access 2
1.2.1. a) Connecting to the console using PuTTy 2
1.2.2. b) Accessing bootloader 3
1.2.3. c) Reading the key-encryption-key (KEK) 3
1.2.4. d) Reading the encrypted AUSKEY 3
1.2.5. e) Decoding the values 4
1.3. 3. Setting up device for Home Assistant control 4
1.3.1. a) Installing a serial port to TCP/IP gateway program. 4
1.3.2. b) Disabling Tuya on device 4
1.3.3. c) (Additional step, not required) Disabling the Tuya customised SSH server and reverting to the "standard" dropbear SSH server on port 22 5
1.4. 4. Installing Home Assistant on host 5
1.5. 5. Using Home Assistant's Zigbee Home Automation to integrate the device 6
1.6. All done! 6

1. Hacking the Silvercrest (Lidl) Smart Home Gateway

1.1. 1. Connecting the device

Main components are detailed here:

```
U2 - A RealTek RTL8196E MIPS System-on-a-chip (SoC) CPU
U3 - A 16MB SPI Flash (GD25Q127)
U5 - A 32MB SDRAM, EM6AA160.
CN1 - A TuYa ZigBee module (TYZS4) which is based on a Silicon Labs ARM-M4
CPU with integrated RF stack. (EFR32MG1B232.)
J1 - A Combined Serial port and ZigBee module ARM debug port. This is not
usually populated. Pinout is:
Pin 1 = Vcc (3.3V) (bottom pin in picture)
Pin 2 = Ground
Pin 3 = U2 Serial TX
Pin 4 = U2 Serial RX
Pin 5 = ZigBee module ARM Debug SWDIO
Pin 6 = ZigBee module ARM Debug SWCLK
```

Serial port parameters are:

```
Electrical: TTL 3.3V
Baud Rate: 38400 bps
Other: 8N1 (8-bit, No Parity, 1 stop bit)
```

To gain access to the Linux console we must connect the 3.3V TTL Serial port to J1. The console Is protected by a password and will be dealt with in a moment.

Our setup looked like this:

1.2. 2. Getting root access

To perform any actions using the Linux console the root password has to be cracked. In order to do that we have to do the following steps:

1.2.1. a) Connecting to the console using PuTTy

First we connect the Gateway to a PC using USB to Serial adapter. The connection configuration is as follows:

```
Serial line: Changes - must be found using device manager
```

```
Speed (baud): 38400
Data bits: 8
Stop bits: 1
Parity: None
Flow control: None
```

1.2.2. b) Accessing bootloader

When the device is connected correctly it has to be restarted to gain access to the bootloader. In order to do that we must turn off the power of the device by unplugging it from the socket. When plugged back in we can see the booting process. Now we must press ESC to land in RealTek bootloader prompt.

We can type ? to see the list of available commands. Available features include the ability to read and write the FLASH, download from a TFTP server into memory, and inspect memory.

The bootloader has a default IP address of 192.168.1.6 and the TFTP server is set to 192.168.1.97.

The bootlog helpfully shows the flash layout as follows:

```
0x000000000000-0x000000020000 : "boot+cfg"
0x000000020000-0x000000200000 : "linux"
0x000000200000-0x000000400000 : "rootfs"
0x000000400000-0x000000420000 : "tuya-label"
0x000000420000-0x000001000000 : "jffs2-fs"
```

1.2.3. c) Reading the key-encryption-key (KEK)

The AUSKEY is itself encrypted by another key that we'll call the KEK. To read the KEK, type the following commands into the bootloader. The first command reads the value from flash into RAM. The next one displays the contents.

```
FLR 80000000 401802 16
DW 80000000 4
```

Make a note of the single line of output. This is your KEK.

1.2.4. d) Reading the encrypted AUSKEY

Once again we input commands which will read the value from flash into RAM and display them.

```
FLR 80000000 402002 32
DW 80000000 8
```

Make a note of the two lines of output. They are your encrypted AUSKEY.

1.2.5. e) Decoding the values

In order to decode the root password we have to use a Python script. It can be found at https://paulbanks.org/download/files/lidl-zigbee/lidl_auskey_decode.py and https://github.com/banksy-git/lidl-gateway-freedom/blob/master/scripts/lidl_auskey_decode.py

Start the python script and, when prompted, paste the lines obtained above to reveal your AUSKEY and root password (we only need password).

Congratulations, if everything worked correctly you are now in possession of the root password to the gateway.

Now most likely you must once again unplug the gateway to reset it. This time let the booting finish. When it is done pressing enter allows logging in. Input root as user and then the password you obtained from the decoding script.

After gaining root access to the system we can see, that we are limited in our actions since the OS is quite bare and there is no distro installed.

1.3. 3. Setting up device for Home Assistant control

Before starting, ensure your device is on your local network. If you're unsure of its IP address, issue an ifconfig command on the serial console. We will use SSH to send files between PC and gateway.

1.3.1. a) Installing a serial port to TCP/IP gateway program.

First we must download the compiled *serialgateway.bin* from https://paulbanks.org/download/files/lidl-zigbee/serialgateway.bin

Now from your host PC run the following:

```
cat serialgateway.bin | ssh -p2333 root@192.168.1.254 "cat
>/tuya/serialgateway"
```

1.3.2. b) Disabling Tuya on device

Next we modify the boot script to disable the Tuya Applications and instead start the serial gatway. To achieve that we execute the following in the connected console:

```
if [ ! -f /tuya/tuya_start.original.sh ]; then cp /tuya/tuya_start.sh
/tuya/tuya_start.original.sh; fi
cat >/tuya/tuya_start.sh <<EOF
#!/bin/sh
/tuya/serialgateway &
EOF
chmod 755 /tuya/serialgateway
```

This will back up the original *tuya_start.sh* script and replace it with a simpler one that simply starts serialgateway.

Once again reboot the device, it should now be controllable by Home Assistant on port 8888.

1.3.3. c) (Additional step, not required) Disabling the Tuya customised SSH server and reverting to the "standard" dropbear SSH server on port 22

Following this step the SSH server will be available only on port 22 and not 2333.

To do it execute the following on the device:

```
if [ ! -f /tuya/ssh_monitor.original.sh ]; then cp /tuya/ssh_monitor.sh
/tuya/ssh_monitor.original.sh; fi
echo "#!/bin/sh" >/tuya/ssh_monitor.sh
```

1.4. 4. Installing Home Assistant on host

There are a few options to install a Home Assistant, we decided to install it with the usage of VirtualBox. First of all it was necessary to download an appropriate file from the official installation guide: https://www.home-assistant.io/installation/linux If someone hasn't got a VirtualBox installed on then it is necessary to download and install it from the official page https://www.virtualbox.org/wiki/Downloads

When we have a VirtualBox installed, then we can create a virtual machine with a Home Assistant. Minimal requirements are:

- 2GB RAM
- 32GB Storage
- 2vCPU

We set our virtual machine following the steps shown below:

1. Create a new virtual machine
2. Select Type "Linux" and Version "Linux 2.6 / 3.x / 4.x (64-bit)"
3. Select "Use an existing virtual hard disk file" - selected the unzipped .vdi file downloaded from an installation guide
4. Edit the "Settings" of the VM and go "System" then "Motherboard" and select "Enable EFI"
5. Then go to "Network" "Adapter 1" choose "Bridged Adapter" and choose your Network adapter
6. Then go to "Audio" and choose "Intel HD Audio" as Audio Controller.

To connect via Ethernet on our university we set the following network settings in VirtualBox

After entering the correct IP in the browser we connected to the Home Assistant:

1.5. 5. Using Home Assistant's Zigbee Home Automation to integrate the device

1. Create a new integration of “Zigbee Home Automation”
2. Select “Enter manually” for the port
3. Pick the EZSP radio type
4. Enter the IP address of your device in this format: socket://ip.address:8888

1.6. All done!

From:
https://dokuwiki.zmitac.aei.polsl.pl/ - **RAU6 Dokuwiki**

Permanent link:
https://dokuwiki.zmitac.aei.polsl.pl/en:edu:zmitac:s2:aih:projekt:studenci:hackingzigbee?rev=1651851361

Last update: **2022/05/06 17:36**