
Philips Hue Bridge v2 hacked (root access)

Posts: 1

1 Topic by pepe2k Today 20:57:11

- pepe2k
- Member
- Offline

- From: **Poland**
- Registered: **2012-02-10**
- Posts: **406**

Topic: Philips Hue Bridge v2 hacked (root access)

Hello!

Philips Hue Bridge v2 owners might be interested in this.
I hope it won't get removed as on reddit...

I saw @wehooper4 work on "jailbreaking" **Philips Hue Bridge v2** and decided to continue, based on what was already discovered. After several hours I was finally able to break inside.

```
BusyBox v1.19.4 (2016-05-10 15:28:31 CEST) built-in shell (ash)
Enter 'help' for a list of built-in commands.

 _    _ _    _ ______   ____       _     _               ___   ___
| |  | | |  | |  ____| |  _ \     (_)   | |             |__ \ / _ \
| |__| | |  | | |__    | |_) |_ __ _  __| | __ _  ___      ) | | | |
|  __  | |  | |  __|   |  _ <| '__| |/ _` |/ _` |/ _ \    / /| | | |
| |  | | |__| | |____  | |_) | |  | | (_| | (_| |  __/   / /_| |_| |
|_|  |_|\____/|______| |____/|_|  |_|\__,_|\__, |\___|  |____(_)___/
                                            __/ |
                                           |___/
-----------------------------------------------------------------------
Version: 33370
-----------------------------------------------------------------------
root@Philips-hue:~#
root@Philips-hue:~# cat /etc/openwrt_release
DISTRIB_ID="QSDK.BSB002"
DISTRIB_RELEASE="1.9"
DISTRIB_REVISION="r40838"
DISTRIB_CODENAME="bsb002"
DISTRIB_TARGET="ar71xx/generic"
DISTRIB_DESCRIPTION="QSDK.BSB002 BSB002 1.9"
DISTRIB_TAINTS="no-all busybox override"
```

TL;DR: you will need solder skills and UART access to the device.

1. Access to U-Boot CLI

First of all, we need to get access to the **U-Boot** command line, to be able to load custom firmware. Because of the boot delay environment variable set to *0 seconds*, U-Boot in Hue Bridge v2 doesn't allow to interrupt booting process at all, so the idea was to break it somehow. The easiest way for that is to temporary disconnect SPI NAND FLASH as it's the one U-Boot is loading kernel from.

There are several ways to make that, including removing whole chip from the board, but thanks to PCB designers, there is a jumper resistor on CS line for the SPI NAND - **R31** (top of the PCB, under main SoC, right bottom corner here). When it's removed, the SoC is not able to enable NAND chip and... U-Boot returns to main loop (CLI):

```
eth1 up
eth0, eth1
Qualcomm Atheros SPI NAND Driver, Version 0.1 (c) 2014  Qualcomm Atheros Inc.
ath_spi_nand_ecc: Couldn't enable internal ECC
Setting 0x181162c0 to 0x3061a100
```

```
Hit any key to stop autoboot:  0

** Device 0 not available
ath>
```

Now we can change bootdelay to something bigger than 0 and save changes (happily this U-Boot version supports writable environment):

```
ath> setenv bootdelay 3
ath> saveenv

Saving Environment to Flash...
Protect off 9F040000 ... 9F04FFFF
Un-Protecting sectors 4..4 in bank 1
Un-Protected 1 sectors
Protect off 9F050000 ... 9F05FFFF
Un-Protecting sectors 5..5 in bank 1
Un-Protected 1 sectors
Erasing Flash... 9F050000 ... 9F05FFFF ...Erasing flash...
First 0x5 last 0x5 sector size 0x10000
   5
Erased 1 sectors
Writing to Flash...  9F050005 ... 9F060000 ...write addr: 9f050000
write addr: 9f040004
done
Protecting sectors 5..5 in bank 1
Protected 1 sectors
Protecting sectors 4..4 in bank 1
Protected 1 sectors
ath>
```

Power down device and solder back jumper resistor to make NAND works again.

2. Dump and extract firmware

In next step I prepared initramfs version of OpenWrt CC image with SPI NAND FLASH support (based on code for GL-AR300M, which is based on same platform), booted it and downloaded dumps from all mtd partitions:

```
root@OpenWrt:/# cat /proc/mtd
dev:    size   erasesize  name
mtd0: 00040000 00010000 "u-boot"
mtd1: 00020000 00010000 "u-boot-env"
mtd2: 00010000 00010000 "reserved"
mtd3: 00010000 00010000 "art"
mtd4: 00400000 00020000 "kernel-0"
mtd5: 02800000 00020000 "root-0"
mtd6: 00400000 00020000 "kernel-1"
mtd7: 02800000 00020000 "root-1"
mtd8: 02800000 00020000 "overlay"
```

As it turned out, rootfs filesystem is **SquashFS** inside **UBI** container (thanks to my colleague @obsy for help with extracting the firmware). Just for reference: extracted root-0 filesystem, from some older version of Hue firmware.

3. How to get root password/access?!

My first idea was simple: extract firmware, change/remove root password, pack firmware and put it back to NAND. But... during looking around, I found script which was called every boot:

```
#!/bin/sh
# Copyright (C) 2015 Philips Lighting

unset UBOOT_SECURITY_STRING
unset SHADOW_SECURITY_STRING

abort() {
    echo -e "$*"
    sleep 1
    exit 1
}

isUBootEnvironmentReady() {
    fw_printenv >/dev/null 2>/dev/null
    return $?
}

updateUBootSecurityString() {
    UBOOT_SECURITY_STRING=`fw_printenv -n security 2>/dev/null`
    return $?
}

updateShadowSecurityString() {
    SHADOW_SECURITY_STRING=`awk -F ':' '/^root:/{print $2}' /etc/shadow`
    return $?
}

escapeStringForSed() {
    echo "$1" | sed -e 's/[\/&]/\\&/g'
}

patchShadowSecurityString() {
    local ESCAPED_SECURITY_STRING=`escapeStringForSed $1`
    sed -i 's/^\(root:\)\([^:]*\)\(.*\)$/\1'${ESCAPED_SECURITY_STRING}'\3/g' /etc/shadow
    return $?
}

syncShadowWithUBootSecurityString() {
    updateUBootSecurityString
    updateShadowSecurityString
    if [ "${SHADOW_SECURITY_STRING}" != "${UBOOT_SECURITY_STRING}" ]; then
        patchShadowSecurityString ${UBOOT_SECURITY_STRING}
    fi
    return $?
```

```
}

if ! isUBootEnvironmentReady; then
    abort "Init in progress: Please try again later..."
fi

if ! syncShadowWithUBootSecurityString; then
    unset UBOOT_SECURITY_STRING
fi
```

The script reads value of *security* U-Boot environment variable, compares it with current root password hash and updates if they are not the same... So, back to U-Boot CLI:

```
ath> setenv security $1\$AeKNkgji\$haI72VcQ8Yi9K5gtL5T1F0
ath> saveenv
Saving Environment to Flash...
Protect off 9F050000 ... 9F05FFFF
Un-Protecting sectors 5..5 in bank 1
Un-Protected 1 sectors
Protect off 9F040000 ... 9F04FFFF
Un-Protecting sectors 4..4 in bank 1
Un-Protected 1 sectors
Erasing Flash... 9F040000 ... 9F04FFFF ...Erasing flash...
First 0x4 last 0x4 sector size 0x10000
Erased 1 sectors
Writing to Flash...  9F040005 ... 9F050000 ...write addr: 9f040000
write addr: 9f050004
done
Protecting sectors 4..4 in bank 1
Protected 1 sectors
Protecting sectors 5..5 in bank 1
Protected 1 sectors
ath> reset
```

At that's all. From now, your **root** password is: **root**

HAPPY HACKING!

U-Boot modification for routers: github.com/pepe2k/u-boot_mod
My blog: www.tech-blog.pl
My photo gallery: galeria.tech-blog.pl

pepe2k's Website

Posts: 1

Pages **1**

You must login or register to post a reply

OpenWrt → Hardware Hacking → Philips Hue Bridge v2 hacked (root access)

Jump to forum:
Hardware Hacking Go