本帖最后由 f541883216 于 2021-3-28 01:27 编辑

本论坛【重磅】全球独家【老骥伏枥-牛年大礼包】黑威联通在线升级的若干问题

前言

本文是【老骥伏枥-狗年大礼包】嵌入式linux逆向工程:手把手教你作黑威联通[1]与【老骥伏枥-原创】制作黑威联通启动盘:进阶篇[2]的续篇。关于破解固件，逆向工程技术的细节在【狗年大礼包】那篇文章中已经详细介绍过了，这里不再赘述。【狗年大礼包】介绍办法属于静态分析的破解，需要使用工具母盘下载固件后打补丁来制作黑威联通启动盘。因此不能实现在线直接升级。也不适合没有linux基础的小白和菜鸟。这次的【牛年大礼包】将介绍如何实现动态破解，即在原始固件启动后，采用挂载“钩子”的办法来实现动态破解。以解决在线固件直接升级的问题。

最近威联通厂家对系统固件频繁升级。终止vQTS系统。并发布了基于公有云的QuTScloud将QNAP NAS的智能操作系统部署于公有云上。在可支持的云端服务中安装QuTScloud操作系统，运行云NAS，解决目前使用中的云端平台无法提供档案管理及弹性共享等需求。透过QuTScloud服务不需要自己处理云端环境复杂的进阶维护，享受轻盈又精省的NAS体验。但这个QuTScloud是需要授权并收费的。而且License费用非常贵，一年的费用足足可以购买自己的私有服务器了。威联通还在不断推出需要License付费的应用，以前免费的应用也逐渐开始变为付费应用。支持这个功能的基础是LicenseCenter QPKG这个应用包。而且QTS也逐渐靠拢以此为基础的License付费应用。为此【牛年大礼包】也做了一些必要的准备。以便于今后可以支持QTS和QuTScloud两种系统。

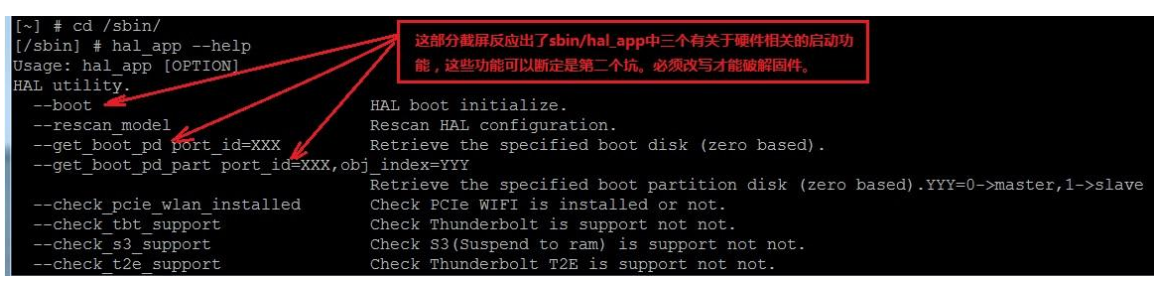
笔者在坛友rmt网友的大力帮助下，先后完成了“破解密文许可证算法及注册机.pdf”，“破解威联通QuTScloud付费License及注册机思路.pdf”，“关于如何实现自授权的方法.pdf”，“关于如何实现通过OpenWRT虚拟路由器授权的方法.pdf”等有关文章。但考虑到这些破解文章的发表不利于厂家的新产品研发，因此暂不发表上述文章。有兴趣研究这些技术的网友，可以联系我们。

需要强调的是，我不是一个黑客。资源来自于威联通官网公布的系统包和发布的固件。参考文献中注明所有引用的出处。本文深入浅出地介绍如何对原始固件启动后安插钩子的办法来实现动态破解的嵌入式linux逆向工程技术。此【牛年大礼包】工具盘的理念和启动引导的基础架构设计均绝对是老骥伏枥本人独创,并合法利用开源项目 (Tiny Core Linux Project) [3] 开发的。 也是老骥伏枥通过 NASYUN 论坛在全球独家首发。此工具盘为的是进行教学， 研究和技术交流。 禁止用于任何商业目的。

第一讲
关于固件的在线升级

自从【狗年大礼包】发表三年以来，广大坛友们勇于实践，使用工具母盘下载固件后打补丁来制作黑威联通启动盘，进行研究和技术交流。大家普遍提出了关于固件的在线升级的问题。由于【狗年大礼包】介绍办法属于静态分析的破解，需要使用工具母盘下载固件后打补丁来制作黑威联通启动盘。因此不能实现在线直接升级。那么如何才能实现在线直接固件升级呢？

让我们先来厘清一些在线直接固件升级到底需要解决什么问题才能实现。固件升级时肯定是固件的代码有了不同程度的更新。这种更改肯定对原有的破解补丁有影响，如果直接固件升级的话，会破坏原有的补丁。退一步而言，就算不升级固件，也不能直接使用原厂固件来覆盖包含有补丁的固件。原因很简单，如果可以直接使用原厂固件而不需要打补丁的话，那就不需要破解了。



【狗年大礼包】中我使用的办法是做一个/sbin/hal_app1模拟/sbin/hal_app三个有关于硬件相关的启动功能，替代/sbin/hal_app的这三个功能。这样做有优点也优缺点。优点是简单易行，缺点是要补丁所有相关的用到这个功能的脚本文件。使用工具母盘的人和仔细读过那篇文章的人都知道，安插补丁需要修改10几个文件。还有就是这样的做法不可以在线固件直接升级。

因为一方面需要安插破解补丁文件，一方面又不能静态插入补丁。解决的思路当然只能是在原始固件启动后，采用挂载“钩子”的办法来实现动态破解。以解决在线固件直接升级的问题。这也是嵌入式linux逆向工程中常用的技术。实现起来也不难。关键的问题是，必须首先完成静态分析的破解才有机会进一步实现动态破解。【狗年大礼包】就是我们这次教程的基础。

这次我们在系统启动之前，不能修改/sbin/hal_app文件了。当然需要改在启动之后才能对原系统固件动手脚。我们可以在启动之后修改/sbin/hal_app。但我再次通过使用IDA Pro工具对/sbin/hal_app仔细分析之后发现。其实不必修改/sbin/hal_app，只需要对model.conf中的启动盘进行仔细配置。它的三个有关于硬件相关的启动功能都是可以正常工作的。

需要关注公众号可查看的内容

威联通系统的model.conf文件采用的是ini格式。ini文件由节、键、值组成。ini格式最早文件是windows的系统配置文件，开始时用于WIN3.1用于管理windows的各项配置。威联通系统利用这样古老的model.conf格式，我猜想可能是为了方便配置时进行手工文本编辑吧。

节
[section]
参数
（键=值）
name=value

因为ini文件可能是项目中共用的，所以使用[Section Name]段名来区分不同用途的参数区。例如：[Boot Enclosure]表示启动盘的参数区；[Boot Disk 1]表示第一个启动盘参数区等等。其中[Boot Enclosure]和[Boot Disk 1]用来表示一个段落。

以前的文章中说过，由于威联通系统的设计采用了硬件绑定的方法。为了适应不同的硬件产品，威联通厂家通过model.conf来对不同的硬件进行配置。让我们以model.conf中关于启动盘配置的项目来说明如下：

[Boot Enclosure]
VENDOR = QNAP
MODEL = BOOT
MAX_DISK_NUM = 1
[Boot Disk 1]
DISK_DRV_TYPE = ATA
DEV_BUS = B00 : D13 : F0
DEV_PORT = 0

其中[Boot Enclosure]节中关键的是MAX_DISK_NUM，它指出有几种可能的启动盘。这里不一定用MAX_DISK_NUM = 1, 我们可以设定为等于2，3等等。

[Boot Disk 1] 节，（如果MAX_DISK_NUM = 2，将需要[Boot Disk 2] 节，以此类推。）DISK_DRV_TYPE部分表示需要设定的启动盘类型，例如：ATA, USB, MMC, VIRTIO, FLASH 等等。

DEV_BUS 部分表示挂接启动盘的PCI总线和设备树编号。其中B，D为十进制编码，F为16进制编码。使用lspci程序读出的数据是十六进制的，需要换算成10进制的对应的B，D，F编码。DEV_PORT部分表示PCI地址和端口号。上述参数必须配置正确，才能保证/sbin/hal_app能正常工作。

配置ATA启动盘比较简单，只要填写对DEV_PORT就可以了。但配置USB启动盘就比较麻烦，因为它涉及到USB接口的挂接有无通过HUB等问题。关于这些在后面的【第四讲：关于USB启动与拔热插配置】中再详细讲解。这里我告诉你一个窍门，配置USB启动盘时最好不要用MAX_DISK_NUM = 1，可以多配置一些，以便你的USB启动盘可以插在多个地方启动。

那么如何检查boot区设定是否正确呢？我在测试时发现了一个小小的技巧。如果boot区设定是正确的，执行get_suid命令时每次获取的数字都是唯一的。如果得到的是随机的数字，那就是boot区设定不正确。这个get_suid命令也很重要，如果不能获取唯一的suid，那么不能支持原厂的同步软件qsync。许可证也要用到suid。

另外，/sbin/hal_app中还有一个蹊跷的细节。它会读取启动盘根目录下的grubenv文件来确定 --get_boot_pd_part port_id=XXX,obj_index=YYY的返回值。这一点我在【狗年大礼包】工具母盘中没有注意到的一个蹊跷的细节。因为采用了替代/sbin/hal_app的这三个功能，所以忽略了这些细节的东西。因为这次将实施动态破解，所以希望改动的东西越少越好。因此再次通过使用IDA Pro工具对/sbin/hal_app仔细分析，才发现了这个问题。这样一来，我们只需要配置model.conf文件和添加grubenv文件，就不需要再对/sbin/hal_app进行任何修改了。与之对应的修改补丁也就都不需要了。

但是为了能够支持各种各样的不同硬件，动态破解补丁仍然是必不可少的。至少model.conf这个配置文件必须在固件启动后，系统建立起来以前，动态地补丁到系统中才行。否则就不可能实现黑威联通。另外还有就是添加必要的网卡驱动模块，系统温度测控Super I/O芯片驱动模块等等。

在讲解详细动态破解补丁之前，让我们再来复习一下【狗年大礼包】中关于威联通启动最小系统内核后的一系列执行动作。这对于我们选择挂载“钩子”的位置甚为重要。

威联通启动最小系统内核后
1.按照/etc/inittab中所述执行以下两个进程
null :: sysinit：/sbin/hwclock - hctosys
:: sysinit：/etc/init.d/rcS

2.根据/etc/init.d/rcS的描述（/etc/rcS_init.d/S20init_check），在/etc/rcS_init.d /下执行每一个名字从S??开始的脚本。它的脚本代码如下截图：

```
[/] # cat /etc/init.d/rcS
#!/bin/sh

. /etc/profile
# Start all init scripts in /etc/init.d
# executing them in numerical order.
#
for i in /etc/rcS_init.d/S??* ;do
     case "$i" in
        *atalk)
            [ -f $i ] && $i start &
            ;;
        *)
            [ -f $i ] && $i start 1>/dev/null 2>&1
            ;;
    esac
done

[ ! -x /sbin/daemon_mgr ] || /sbin/daemon_mgr
[/] #
```

3.执行/etc/rcS_init.d/S20init_check并且开始实施安装各种设备。

从上述的分析中可以看出，我们可以有几种挂载“钩子”的位置：
a. 添加一个/etc/rcS_init.d/S10hook_init_before_S20文件，让它在开始执行S20init_check之前做些必要的补丁任务。
b. 添加一个/etc/rcS_init.d/S21hook_init_after_S20文件，让它在开始执行S20init_check之后做些必要的补丁任务。
c. 因为S20init_check开始实施安装各种设备，而有些任务需要插入S20init_check之中进行，因此我们可以用sed命令在S20init_check脚本中插入钩子。

本次的【牛年大礼包】就是采用sed命令在S20init_check脚本中插入钩子办法。在linux启动的grub.cfg文件中，使用initrd ($bootdev,2)/boot/initrd.boot ($root)/patch.gz的方法，在initrd.boot之后再次加载patch.gz补丁，覆盖掉/etc/model.conf文件和覆盖掉/sbin/init执行脚本，并在init中安插laojifuli_patch钩子，这是一个脚本文件，放在patch.gz补丁中，大家可以根据自己的需要把任务添加到它里面去。例如：添加必要的网卡驱动模块，系统温度测控Super I/O芯片驱动模块等等。具体源代码请参见/sbin/init文件。我在【牛年大礼包】中，已经添加了一些必要的虚拟机用的网卡驱动模块。最新的版本中，网卡驱动模块/etc/init.d/load_lan_module.sh脚本来管理。加载网卡驱动模块就变得非常容易了。只需要把相应的.ko文件拷贝到/lib/modules/4.14.24-qnap目录中即可。

由于NASYUN的篇幅限制，请看楼下，第二讲 【关于序列号与洗白】，精彩继续！

f541883216 发表于 2021-2-13 08:32 沙发

第二讲及第三讲

本帖最后由 f541883216 于 2021-2-22 07:48 编辑

第二讲
关于序列号与洗白

【狗年大礼包】的破解中是不包括序列号的，其实这个序列号与洗白系统也没有什么直接的关系。但是没有序列号，大家总是认为没有洗白系统。

威联通系列号好像没有什么用途。它是由十位的字符和数字组成。例如：Q152A00888。我去查了一些白威联通的系列号，前面4位好像是机型。第5位好像是产品批次。第6-10位好像是序号。

威联通厂家通过model.conf来配置不同机型的序列号。配置项目是：

[System Enclosure]
BOARD_SN_DEVICE = NET　把序列号存放到网卡的eeprom中

在[System Enclosure]节中使用关键词BOARD_SN_DEVICE指出序列号存放的位置。例如：NET，VPD，VPD:BP，VPD:MP，USBDOM，KVM，BOOT等等。总之是一些威联通厂家规定的特殊地方来存放序列号。

在上述的这些位置中，我们可以使用IDA Pro工具去分析get_hwsn这个二进制文件，来确定如何将序列号写入。通过分析发现，有很多办法存放序列号。下面介绍几种简单的：

本帖隐藏的内容

第一种：使用NET:X把序列号存放到网卡的eeprom某个特定位置。X是网卡的编号。例如eth0,eth1等等。

首先你需要使用 /usr/sbin/ethtool 这个工具检查一下网卡的eeprom是否容许改写。例如查看虚拟机的虚拟网卡就可以使用如下命令：

```
[~] # /usr/sbin/ethtool -i eth1
driver: virtio_net
version: 1.0.0
firmware-version:
expansion-rom-version:
bus-info: 0000:00:03.0
supports-statistics: no
supports-test: no
supports-eeprom-access: no
supports-register-dump: no
supports-priv-flags: no
```

一般而言，虚拟机的虚拟网卡都不容许访问eeprom。但实体机的实体网卡一般都是可以访问的。
威联通规定了两个位置存放序列号到网卡，0x38 和 0xC00 视网卡的类型而定。一般是写在0xC00 这个位置。可以用如下命令写入序列号：

```
[~] # /usr/sbin/ethtool  -E eth1 offset 0xC00 length 0x17 value  “SN : Q123A12345”
```

第二种：使用USBDOM把序列号存放到启动盘的第五分区的usbdom.conf文件中。

这种方法我认为是最简单方便的方法。在启动盘的第五分区创建一个usbdom.conf文件。它的内容如下：

```
[main]
serial=Q123A12345
```

然后修改model.conf配置为：

```
[System Enclosure]
BOARD_SN_DEVICE = USBDOM
```

这种办法虚拟机和实体机都适用。

第三种：使用BOOT把序列号存放到启动盘的某个特定位置。

当boot区设定正确后，就可以把序列号写入到启动盘的某一个分区中去了。写法如下：

BOARD_SN_DEVICE = BOOT: 偏移量

表示在BOOT设备的偏移量处读取序列号，BOOT即启动磁盘，如果是BOOT1，BOOT2，BOOT3之类就是BOOT磁盘的第几个分区的偏移。为了确保写入的数据不被覆盖掉，我的【牛年大礼包】的第一个启动分区为64M。建议写入到比较靠后边的地方。按字节计算是64x1024x1024=67,108,864。因为威联通序列号是10位，为了方便记忆，我会将它写到偏移量=67,000,000 的位置。这是一个非常靠后的地方，一般情况下我们不会将启动分区的数据写得如此满，因此是一个比较安全的位置。写入方法如下：

```
dd if=/dev/zero of=/dev/sda1 seek=67000000  bs=1 count=20
echo -n "Q152A00888" | dd of=/dev/sda1 seek=67000000  bs=1
```

其实配置序列号很简单。我在【牛年大礼包】工具盘中专门为大家写了一个set_hwsn的工具。大家可以随心所欲地配置自己喜欢的序列号。只要合乎由十位的字符和数字组成的厂家的规则就可以。例如Q666A88888，“666大顺88888发发发”。



给个截图，这是我在蜗牛星际矿渣机的TS-453B系统上生成的序列号。

第三讲
关于系统温度与Super I/O芯片

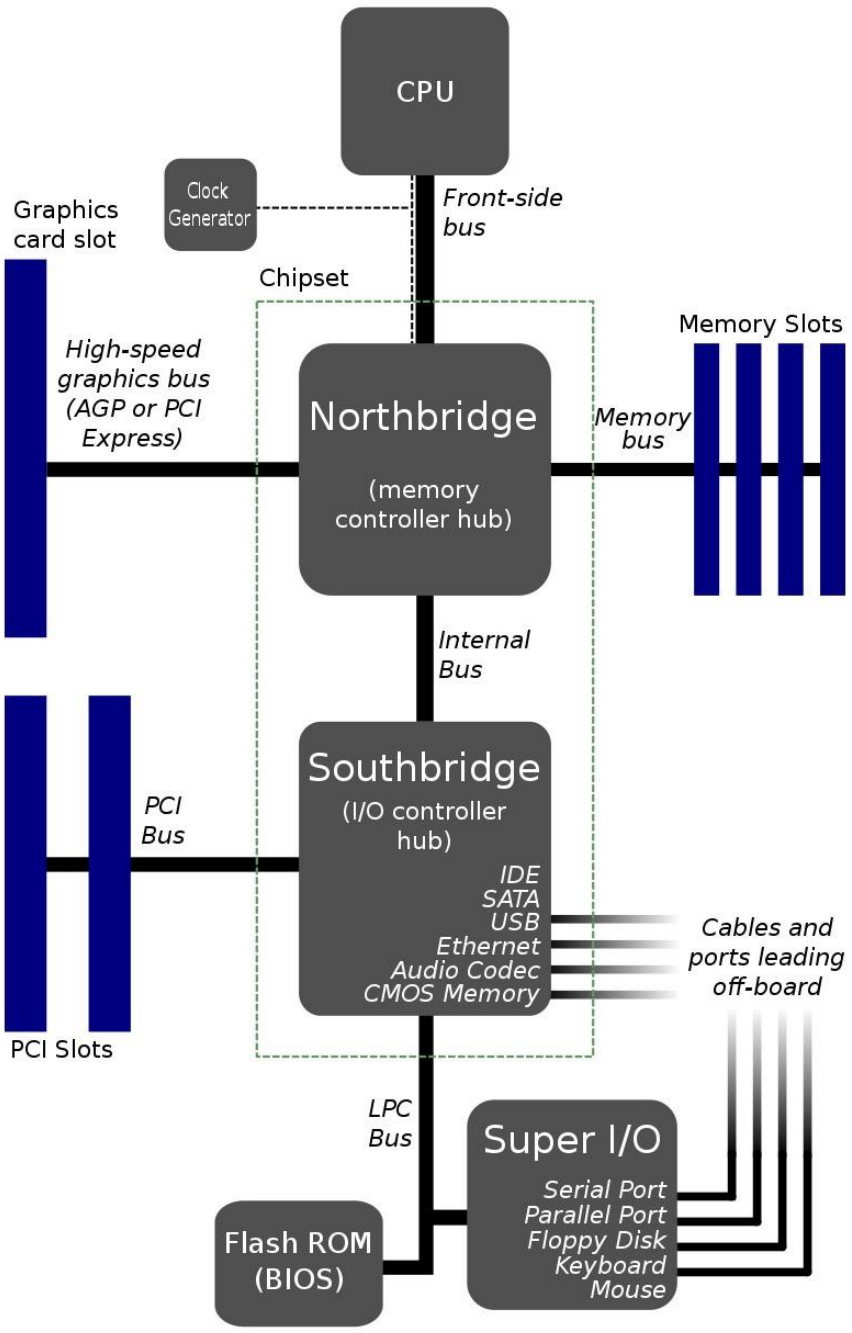
威联通厂家的系统温度，硬盘温度，风扇等等都是使用Super I/O芯片以及相关的linux驱动模块实现的。

为了让大家更了解SIO，接下来我要介绍一些关于Super I/O芯片的知识。

Super I/O 是一种I/O芯片，于1980年代后期被使用于个人电脑的主板内，起初是以界面卡的形式出现，后来直接内建于主板上。一个Super I/O芯片包含了若干各式低带宽设备的界面。其功能通常包含以下项目:

软盘控制器，并行口 (通常使用于打印机)，串行口，PS/2接口键盘与鼠标界面。温度感测器与风扇速度监测器，风扇速度控制。Super I/O芯片还可能包含其他界面，例如游戏接口或红外线接口。 因为已经将多个功能整合于一颗芯片内，降低了主板上所使用的零件数，也因此降低了产品的成本。有些Super I/O芯片还支援机壳开启侦测。

最早期的Super I/O芯片透过ISA总线与中央处理器沟通。现在的Super I/O芯片则是使用LPC总线，取代原先的ISA总线，与中央处理器沟通。通常透过主板上的南桥芯片上的LPC总线界面来完成。



Super I/O芯片有很多厂家生产，型号也五花八门。但目前主流的Super I/O芯片有两大类。Intel 的8XXX及其兼容系列，Nuvoton 的nct6XXX 及其兼容系列。威联通厂家的固件基本上是在内核中build in了IT8528的驱动。在model.conf中的配置项目是：

[System Enclosure]
SIO_DEVICE = IT8528

如果你的硬件也是使用这个系列的Super I/O芯片，也不可以直接利用内核中的内置的EC驱动。 因为威联通的内核内置EC驱动是特殊定制的。所以如果你需要系统温度，硬盘温度，风扇等等信息的话，就要自己补充uper I/O芯片的驱动模块。但虚拟机一般不会包含Super I/O芯片的仿真。如果你将固件安装到虚拟机上，则应该从model.conf文件中的

删除配置项目：

[System FAN]
[CPU FAN]

[System FAN]节以及[CPU FAN]节中的所有内容。

对于实体机硬件，你可以通过BIOS设定中的Advancement菜单查看Super I/O芯片的型号。如果你的硬件使用的是Intel 的8XXX及其兼容系列，你就要补充it87.ko模块。并把mode.conf里面所有的ec改成sio。让系统通过Super I/O芯片获取系统温度，硬盘温度，风扇等等。这样的话应该是可以显示出某些系统温度的。至少有一部分是可以的。但如果你的硬件是Nuvoton 的nct6XXX 及其兼容系列，就需要添加相关nct6XXX.ko的驱动模块。

如果你的硬件是非主流Super I/O芯片的型号，你就要添加相关的驱动模块。

添加相关的驱动模块你就需要下载威联通的工具链和linux系统的内核代码包资源的网址是：

网址： https://sourceforge.net/projects ... NAS%20GPL%20Source/

大家可以去下载它的工具链。编译内核或各种自己需要的外挂模块。具体方法请参阅它的用户手册。目前威联通官网发布的固件4.5.2是使用Linux的4.14.24-qnap内核签名。工具链编译器是toolchain config: [gcc-4.9.2 binutils-2.25 glibc-2.21]。如果你自己编译内核有困难。可以到网上找找有没有适合版本的模块。

编译驱动模块并不困难，只要有耐心一定会成功的。Super I/O芯片的驱动，建议编译成模块，不要编译为内核嵌入。例如：Nuvoton 的nct6XXX 及其兼容系列的驱动模块，可以编译成nct6xxx.ko模块，然后通过laojifuli_patch钩子动态加载到系统中。就算是在线固件升级，也不会有影响。

另外，就是要调整mode.conf里面的设置。

```
[System Enclosure]
SYSTEM_TEMP_UNIT=SIO:1
[System FAN]
FAN_UNIT = SIO
FAN_1=I2
[CPU FAN]
FAN_UNIT = SIO
FAN_1=I3
```

系统温度 取SIO:1表示取nct6xxx.ko的第一温度。下面是风扇 取第2 以及3位置的风扇转速等等。这些参数都要根据你的硬件具体是如何与Super I/O芯片的驱动对应有关。还有在编译驱动模块时的设定来调整才行。

加载驱动成功后，不论什么硬件都会在/sys/class/hwmon/hwmonX目录里得到所有系统温度，硬盘温度，风扇等等信息的数据文件。但威联通系统只会读取/sys/class/hwmon/hwmon0目录里的数据。有两种办法解决这个问题，第一种是修改libuLinux_hal.so库，把它调整到你的目录，但这涉及到hex文件的编辑修改。第二种是，修改/etc/init.d/init_check.sh文件，去掉原来加载的有关hwmon模块。（ coretemp.ko和k10temp.ko ）这样你加载的模块数据就会在hwmon0目录中了。

Super I/O芯片与驱动都是低速的简单模块。也都很小，这部分不是威联通QTS系统破解的重点。

由于NASYUN的篇幅限制，请看楼下，第四讲 【关于USB启动与热拔插配置】，精彩继续！

点评 回复 支持 反对 评分 举报

f541883216 发表于 2021-2-13 08:32 板凳

第四讲 关于USB启动与热拔插配置

本帖最后由 f541883216 于 2021-2-22 07:50 编辑

第四讲
关于USB启动与热拔插配置

威联通的USB启动与热拔插配置也是在mode.conf里面设置的。首先你要知道有多少个USB口配置。这个配置在[Usb Enclosure]节中描述。例如：

```
[Usb Enclosure]
VENDOR = QNAP
MODEL = USB
MAX_PORT_NUM = 8
USB3_PORT_BITMAP = 0x1FE
```

参数MAX_PORT_NUM = 8表示有八个USB口要配置。参数USB3_PORT_BITMAP = 0x1FE表示如果插入USB3.0的外设。使用“0x1FE”识别符来鉴别。

USB启动与热拔插配置比较复杂。主要原因是硬件设备五花八门，USB设备还会有多层次的HUB。另外威联通的mode.conf设置写法也分为USB 2.0和 USB 3.0 等不同。这里我仅仅以某个实例加以说明，以便于大家理解：

对于不同的硬件设备，接下来在不插入任何USB设备的情况下，最好在启动的威联通QTS系统环境中查看。要使用lsusb命令检查一下机器硬件的USB配置情况。我以蜗牛星际矿渣机“A款单82583V网卡”主板为例做个说明：

```
Bus 001 Device 002: ID 05e3:0608 Genesys Logic, Inc. USB-2.0 4-Port HUB
Bus 001 Device 001: ID 1d6b:0002 Linux Foundation 2.0 root hub
Bus 002 Device 001: ID 1d6b:0003 Linux Foundation 3.0 root hub
```

这个硬件的Bus 001 Device 001是USB-2.0 的root hub如果你把一个USB 2.0的外设插入机器的A款机箱左后下方口（有蓝色标记的USB-3.0端口），这个root hub就会激活1-1。

这个硬件的Bus 002 Device 001是USB-3.0 的root hub如果你把一个USB 3.0的外设插入机器的A款机箱左后下方口，这个root hub就会激活为2-1。

这个硬件的Bus 001 Device 002上内置接有一个USB-2.0 的4-Port HUB。蜗牛矿渣机箱前后共有6个USB口，机箱内部还有2个。一共是8个USB接口。但只有在机箱左后下方的USB口（代蓝色标记的那个口）支持usb3.0。它的DEV_BUS = B00 : D20 : F0 需要填写到对应的[Usb Port 1 - 8]节。如果你把一个USB 2.0的外设插入机器其它USB口，该口如果通过了内置的HUB，就会激活为1-2.X。其中的X是表示HUB上的某个口。如果没有通过内置的HUB，则会激活root hub的1-X。

接下来让我们用dmesg命令查查插入每个USB接口后的编号标识。请看如下给出的6个外部的USB接口信息：

[6118.042255] usb 1-2.3: new high-speed USB device number 7 using xhci_hcd (A款机箱左前USB口）
[6601.307307] usb 1-2.2: new high-speed USB device number 9 using xhci_hcd (A款机箱右前USB口）
[6706.240317] usb 1-4: new high-speed USB device number 10 using xhci_hcd （A款机箱左后上方）
[6925.795334] usb 1-1: new high-speed USB device number 11 using xhci_hcd （A款机箱左后下方，支持usb3.0）
[7101.987358] usb 1-2.1: new high-speed USB device number 12 using xhci_hcd （A款机箱右后上方）
[7277.858371] usb 1-3: new high-speed USB device number 13 using xhci_hcd （A款机箱右后下方）

为了获取这些编号，必须每个USB口都插入一遍，才能得到这些信息。对于需要填写到对应的[Usb Port 1 – 8]节，你也可能不需要填写这么多的[Usb Port X]节。但是除了填写DEV_BUS之外，对于USB口还需要填写以上的三项或两项信息。另外设置写法也分为USB 2.0和 USB 3.0 。因此你必须仔细考量，你用的USB存储器与你的硬件USB接口两个方面综合来决定。威联通的硬件设备是固定的，它会配置好mode.conf文件。但对于其他五花八门的硬件，配置起来就比较麻烦了。但只要你仔细阅读本文，还是可以配置成功的。

直接连到root hub的USB口只需要填写两项信息。通过内置HUB的USB口则需要填写三项信息。

上述信息是我用一个USB2.0的U盘，插遍矿渣机外部的6个USB口得到的信息。可以看出，有些USB口是直接连到root hub的，而另一些是通过内置的USB-2.0 的4-Port HUB的。

本帖隐藏的内容

如果使用 SUB 2.0 的写法：（ 上述2.0 root hub 接在 BUS 001 上）

IN_HUB = 1　根 HUB（Device 001: ID 1d6b:0002 Linux Foundation 2.0 root hub）
HUB_PORT = 2　HUB 端口（Device 002: ID 05e3:0608 Genesys Logic, Inc. USB-2.0 4-Port）
DEV_PORT = 1 具体你的 SUB 插在那个USB-2.0 4-Port HUB的端口上。没有没有通过内置的HUB，则不需填写。

如果使用 SUB 3.0 的写法：（ 上述3.0 root hub 接在 BUS 002 上）

DEV_PORT_SSP = 1 这个是总线BUS 的编号。1指的是最前边的编号。（USB 2.0设备插在3.0 root hub上的情况。相当于SUB 2.0 的写法的 IN_HUB）
DEV_PORT_NS = 2　这个2指的是中间的编号。（如果没有第三项，它就是最后一项。相当于SUB 2.0 的写法的HUB_PORT）
DEV_PORT_SS = 3　这个3指的是最后边的编号。如果没有就不需要填写。（相当于SUB 2.0 的写法的DEV_PORT）

对于[Usb Port X] 节，设置写法也分为USB 2.0和 USB 3.0两种配置方法，我在这里仅仅以USB 3.0为例。

[Usb Port X]　　其中的X应为1 – 8。如果你的机器只有3个口，则为 1 – 3。
DEV_BUS = B00 : D20 : F0　　lspci读出的设备总线。
DEV_PORT_SSP = X
DEV_PORT_NS = X
DEV_PORT_SS = X

DEV_PORT_SSP = X 这个X指的是最前边的Bus编号。
DEV_PORT_NS = X　这个X指的是中间的Device编号。（如果没有通过内置HUB，就不需要第三项信息了。）
DEV_PORT_SS = X　这个X指的是最后边HUB口的编号。

例如，对于A款机箱左前USB口，把它配置为[Usb Port 1] 节的情况如下：

根据信息：[6118.042255] usb 1-2.3: new high-speed USB device number 7 using xhci_hcd (A款机箱左前USB口）

[Usb Port 1]
DEV_BUS = B00 : D20 : F0
DEV_PORT_SSP = 1
DEV_PORT_NS = 2
DEV_PORT_SS = 3

如果这些信息都填写正确，就可以实现USB启动与热拔插配置。关于USB的配置，不论是启动盘，还是热拔插的USB外设信息都是一致的。有些时候如果使用 SUB 3.0 的写法配置不能工作，可以考虑改用UB 2.0 的写法配置。这一点在配置USB启动盘时非常重要。我的窍门是，同时用两种写法配置启动盘。谨防USB启动盘配置不能正常工作。

对于4.4.1以上系统想要实现usb启动和自适应即插即用，需要满足两个条件 第一在model里关于引导的写法一定要正确。通过内置HUB的USB口则需要精确到port。第二model里usb的写法也要正确，usb3.0 usb2.0 两种写法各不同。

由于NASYUN的篇幅限制，请看楼下，第五讲 【工具盘使用与实践范例】，精彩继续！

f541883216 发表于 2021-2-13 08:33 地板

第五讲 【工具盘使用与实践范例】

本帖最后由 f541883216 于 2021-2-18 09:40 编辑

第五讲
工具盘使用与实践范例

为了便于大家学习掌握嵌入式linux逆向工程知识，【牛年大礼包】将为大家提供一个全新设计的工具盘。此次的工具盘以iso格式提供。该盘中不包含任何原厂的固件代码。该盘支持以三种方式下载的原厂固件自动破解并创建启动盘。同时支持UEFI与BIOS Legacy两种启动方式。使用起来比以前的工具母盘简单而方便。适合没有linux基础的小白和菜鸟。该盘的设计理念，系统结构，所包含的程序代码均为老骥伏枥本人独家原创已及合法使用开源的TinyCore系统。全部是拥有自主知识产权的本人作品。基本上实现了一键安装。请坛友们遵守 GNU General Public License， 未经授权禁止用于商业用途。 该演示用工具盘仅仅为的是进行教学， 研究和技术交流。 禁止用于任何商业目的。

本帖隐藏的内容

该盘的提取地址是: https://pan.baidu.com/s/1CLxe1siTwiFFO5to3HERpg
提取密码: 7ig2



接下来我首先要介绍【牛年大礼包】工具盘的设计理念。这次的理念不同于【狗年大礼包】提供的工具母盘。我将创建黑联通系统启动盘分为三个独立的步骤：

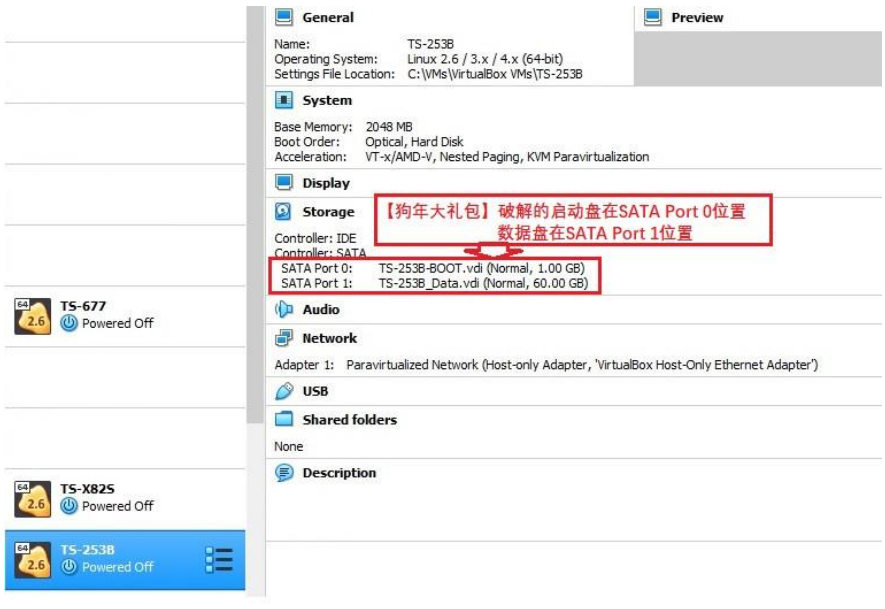
第一步：使用工具盘获取威联通系统的固件。支持选择任意版本的固件。这次的工具盘将支持以三种方式来获取固件。1.直接从官网URL获取原厂的系统文件并提取固件。2.从已经下载的zip文件中提取固件。3.从已经使用【狗年大礼包】破解的启动盘中提取破解后的固件。本次的【牛年大礼包】工具盘也支持以前使用【狗年大礼包】破解的系统解决在线固件升级。

第二步：由于不再需要制作补丁，但因为威联通系统是“硬件绑定”的，因此仅仅需要手工编辑一下mode.conf文件来与你的硬件相对应就可以了。你可能会说：“为什么要手工编辑mode.conf文件？不能自动生成吗？”关于这个问题，我认为手工编辑是最好的选择。因为各种机型的硬件千差万别，就算是相同的机型，硬盘的接线连到不同的SATA端口，都会直接对mode.conf文件产生影响。目前没有更好的办法制作黑威联通启动盘。

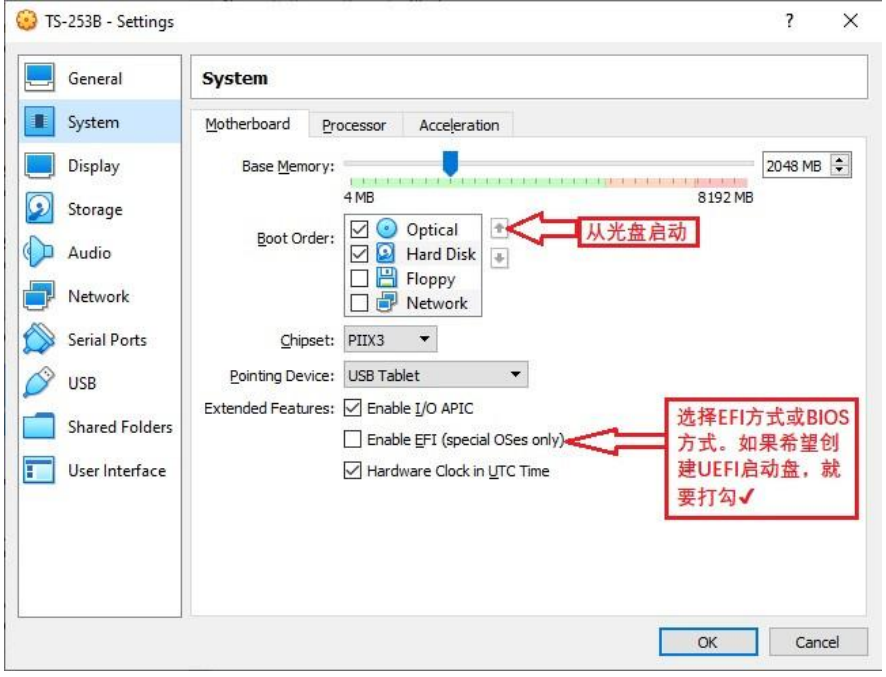
第三步：创建黑威联通启动盘。使用【牛年大礼包】工具盘创建的启动盘不仅仅是启动盘，它还是一个工具母盘。它将继承工具盘的功能。能以此盘为基础“自举式”再创建黑威联通启动盘。因此，一旦你使用【牛年大礼包】工具盘创建黑威联通启动盘成功。工具盘就再也不需要了，可以把它扔掉了。

接下来我将要以VirtualBox虚拟机为例，介绍几种具体的实践范例。

场景1：我已经使用【狗年大礼包】的方法破解了TS-253B并且能正常工作，但不能在线固件升级。



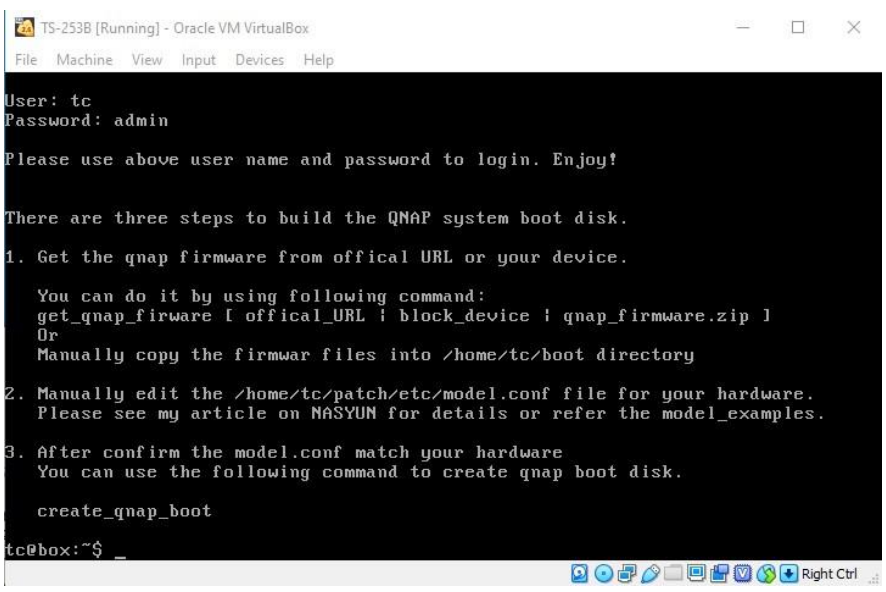
将【牛年大礼包】工具盘Ox_install_UEFI_BIOS.iso的iso文件添加到虚拟机的IDE口，并设置为从CDROM启动。请看如下截屏：



设置完成以后，就可以启动虚机了。请看如下截屏：



此时只能选择“创建威联通启动盘”，输入用户名：admin和密码：admin。然后会进入Tiny Core的控制台界面。此时我已经为你打开了SSHD服务，容许你使用PuTTY SSH Client方便连接操作。屏幕上显示有英文的操作说明，为什么不用汉语的呢？因为希望这个工具盘尽量小，没有安装汉字字库。请看如下截屏：



接下来，就可以按照【牛年大礼包】工具盘的设计理念的三个步骤进行操作。

注意：所有输入的创建启动盘的命令，必须在tc用户当前的默认目录下：tc@box:~$ 如果你切换到其他目录中了，可以使用命令：cd <回车> 切换回默认目录。

第一步输入命令。从【狗年大礼包】的启动盘提取固件。因为启动盘在SATA Port 0的位置上。所以它的设备名应该是：/dev/sda。提取的命令是：

tc@box:~$ get_qnap_firmware /dev/sda

提取成功后，做第二步操作。手工编辑/home/tc/patch/etc/model.conf文件。因为以前的系统是可以运行的黑威联通系统，因此这里基本上不需要做任何model.conf的更改。但为了安全起见。还是建议你使用如下命令核对一下系统配置情况。

```
tc@box:~$ lspci
00:00.0 Host bridge: Intel Corporation 440FX - 82441FX PMC [Natoma] (rev 02)
00:01.0 ISA bridge: Intel Corporation 82371SB PIIX3 ISA [Natoma/Triton II]
00:01.1 IDE interface: Intel Corporation 82371AB/EB/MB PIIX4 IDE (rev 01)
00:02.0 VGA compatible controller: InnoTek Systemberatung GmbH VirtualBox Graphics Adapter
00:03.0 Ethernet controller: Red Hat, Inc Virtio network device
00:04.0 System peripheral: InnoTek Systemberatung GmbH VirtualBox Guest Service
00:06.0 USB controller: Apple Inc. KeyLargo/Intrepid USB
00:07.0 Bridge: Intel Corporation 82371AB/EB/MB PIIX4 ACPI (rev 08)
00:0b.0 USB controller: Intel Corporation 82801FB/FBM/FR/FW/FRW (ICH6 Family) USB2 EHCI Controller
```

00:0d.0 SATA controller: Intel Corporation 82801HM/HEM (ICH8M/ICH8M-E) SATA Controller [AHCI mode] (rev 02)

tc@box:~$ vi /home/tc/patch/etc/model.conf

确认其中的启动盘配置信息是正确的：

[Boot Enclosure]
VENDOR = QNAP
MODEL = BOOT
MAX_DISK_NUM = 1
[Boot Disk 1]
DISK_DRV_TYPE = ATA
DEV_BUS = B00 : D13 : F0
DEV_PORT = 0

红色的部分必须正确，否确会启动失败。

一切确认无误后，做第三步操作。输入命令:

tc@box:~$ create_qnap_boot

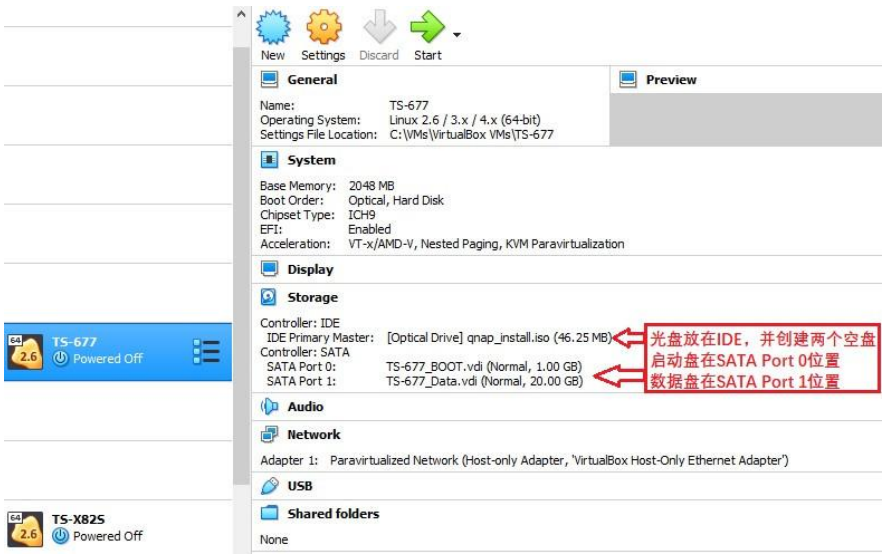
根据提示选择启动盘的设备名，例如：现在的虚机中，只有/dev/sda（旧启动盘）和/dev/sdb（旧数据盘）两个设备。如果你不想覆盖掉旧启动盘，也不想破坏旧数据盘。那么你应该在SATA Port 2或3的等后续位置上再加一个盘。如果用作启动盘，尺寸有1G就可以。那么就会有/dev/sdc设备。你也可以添加多个设备，用来创建多个启动盘。

创建启动盘完成后，就可以关机。然后使用新创建的启动盘了。

注意：因为这个启动盘创建的时候，model.conf文件配置为SATA Port 0的位置上。因此这个盘必须放回在SATA Port 0的位置系统才能正常启动。

场景2：我想从官网下载一个TS-X77_20200214-4.4.1.1216.zip固件，用VirtualBox虚拟机新建一个黑威联通。

虚拟机的配置截图如下：



设置为光盘启动。启动后进入Tiny Core的控制台界面的截图与前面的截图是一样的。这只是提取固件的命令有所不同而已。直接从官网下载并提取固件的命令是：

tc@box:~$ get_qnap_firmware https://download.qnap.com/Storag ... 0214-4.4.1.1216.zip

当然也可以预先从官网下载TS-X77_20200214-4.4.1.1216.zip文件，然后把它拷贝到/home/tc目录下。使用如下命令提取固件。

tc@box:~$ get_qnap_firmware TS-X77_20200214-4.4.1.1216.zip

总之，这次的【牛年大礼包】支持以三种方式下载的原厂固件自动破解并创建启动盘。非常方便给没有经验的小白和菜鸟使用。

提取到固件后，就要做第二步操作。手工编辑/home/tc/patch/etc/model.conf文件。需要配置的信息包括，**启动盘位置，数据盘的个数和位置，USB的个数和位置，系统温度控制器**，等等。如果实在不知道如何填写，只能根据你的经验，或者删去，或者保持原样。

注意：model.conf文件中的配置信息，是不可少的。上述“红色”的部分。都必须根据具体硬件填写正确。才能保证威联通系统工作正常。我在【牛年大礼包】工具盘中给了几个范例。提供参考，如何配置model.conf文件。

对于小白和菜鸟而言，这一步是能否成功的关键。对于已经熟悉【狗年大礼包】的坛友们而言，这一步就是小菜一碟。但这里我还是要简单地讲一讲。

在工具盘有一个/home/tc/model_expamles目录。里面会存放从获取的固件中获得的各种model_*.conf文件的实例。也有一些我为大家创建的实例。厂家的实例有他们的命名规则，我们无从可知。但我的实例将使用如下的命名规则。

例如：model_TS-453B_VirtualBox_virtio-net_SATA.conf.
第一段：model代表黑威联通的硬件配置文件。
第二段：TS-453B代表黑威联通TS-453B系统的硬件配置文件
第三段：VirtualBox代表黑威联通给VirtualBox虚拟机的配置文件
第四段：virtio-net代表黑威联通在VirtualBox虚拟机上用virtio-net网卡配置
第五段：SATA代表黑威联通启动盘是SATA Port 0位置的硬件配置文件

有了这个model.conf文件实例，就可以给虚拟机直接创建TS-453B系统的启动盘了。仅仅需要把它拷贝到/home/tc/patch/etc/model.conf中覆盖掉原来的model.conf就可以了。简单吧。

今后，大家可以按照命名规则直接交换各种各样的model_*.conf文件。方便大家创建各种各样版本和系统的启动盘了。如果小白和菜鸟实在不会做model.conf文件，可以找有经验的人帮忙，只要拿到“硬件绑定”关联的model.conf文件。就可以创建黑威联通启动盘。

如果没有实例的model.conf也没有关系。只要按照前面原理中讲的规则，修改/home/tc/model_expamles目录中的某个文件，使之适合你的硬件，再把它拷贝到/home/tc/patch/etc/model.conf中覆盖掉原来的model.conf就可以了。

以前的【狗年大礼包】需要制作补丁，不适合小白和菜鸟。现在的【牛年大礼包】工具盘，仅仅需要编辑一个“硬件绑定”关联的model.conf文件就可以创建黑威联通启动盘了。大大简化了操作。使得小白和菜鸟都可以制作黑威联通启动盘。福利还不仅如此，此次的启动盘还可以在线固件升级和洗白。简直就是可以为任何硬件制作一个“白威联通”系统。不过老骥伏枥再次提醒大家：如果你喜欢威联通系统，请购买正版的“白威联通”系统。不要使用教学用破解版的威联通系统。

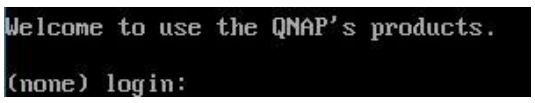
接下来的第三步，与前面所说是一样的。输入命令:

tc@box:~$ create_qnap_boot

根据提示选择启动盘的设备名，因为启动盘在SATA Port 0的位置上。所以它的设备名应该是：/dev/sda。

创建完成后，就可以关机。重新配置虚拟机，拿掉IDE上的iso工具盘。使用新创建的启动盘试试吧。

如果看到如下这个画面出现在屏幕上。那就是启动出现错误了。



出现这种错误的原因一定是model.conf配置不正确的问题。没有关系，工具盘提供了两个命令让你可以快速重新再试model.conf配置。重启系统，再次进入Tiny Core的控制台界面。

使用命令：

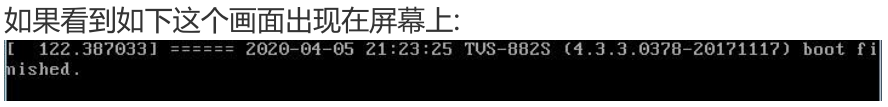
tc@box:~$ patch_extract

再次做第二步操作。手工编辑./patch/etc/model.conf文件，调整配置。
调整完成后，使用命令：

tc@box:~$ patch_install

重启系统再试，不断调整配置，直至成功。

如果看到如下这个画面出现在屏幕上:



或者是如下这个画面:



那就是启动成功了。就可以进行辅助配置安装了。

这次的【牛年大礼包】工具盘，提供了丰富的工具，帮助大家创建启动盘。很高兴能与大家分享此次的成果。希望大家喜欢。

场景3：我已经使用【牛年大礼包】工具盘成功地创建了黑威联通。如何设定序列号。

注意：强烈建议不要这样做。特别是在还没有成功地创建黑威联通系统之前不要设定序列号。

启动进入Tiny Core的控制台界面。在当前目录下，有一个QNAP_HWSN的文件。里面是默认的“Q000A00000”序列号。使用vi编辑它为一个合乎厂家的规则的由十位的字符和数字组成的序列号。

例如Q666A88888，“666大顺88888发发发”。

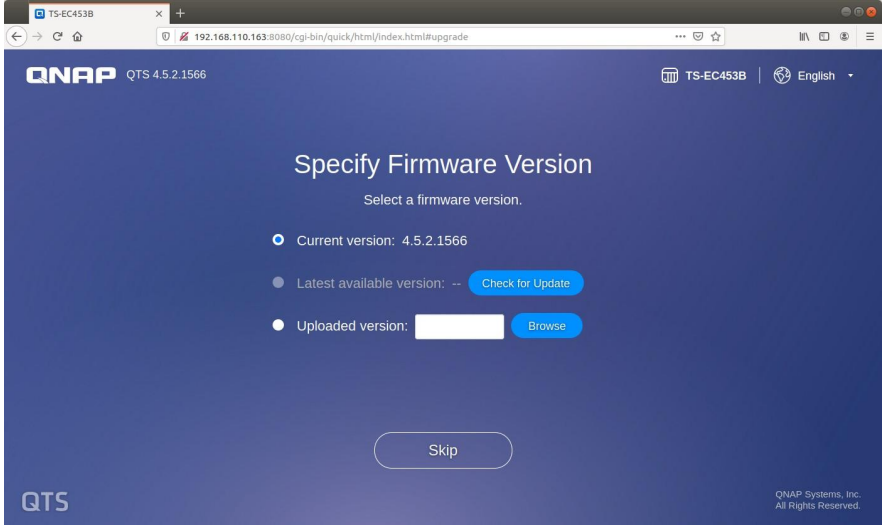
然后执行命令：

tc@box:~$ set_hwsn

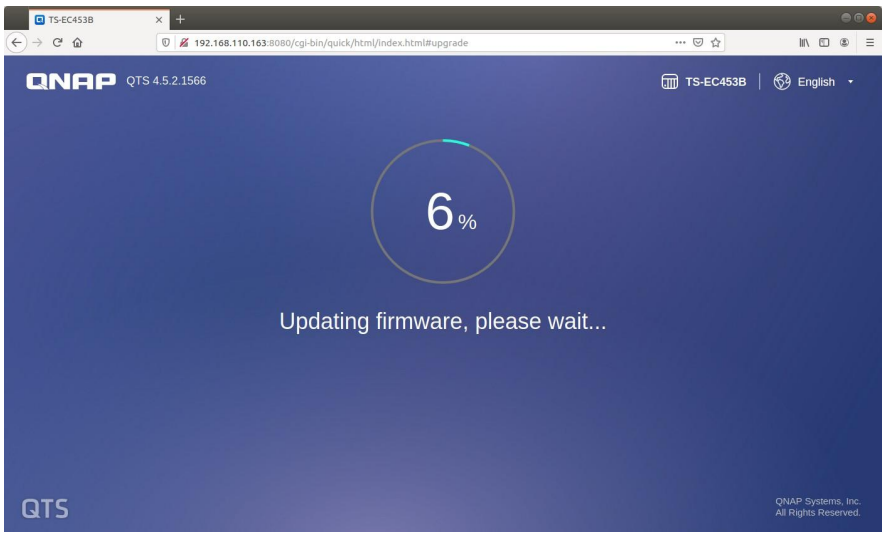
成功后，重新启动系统。你就可以看到序列号了。

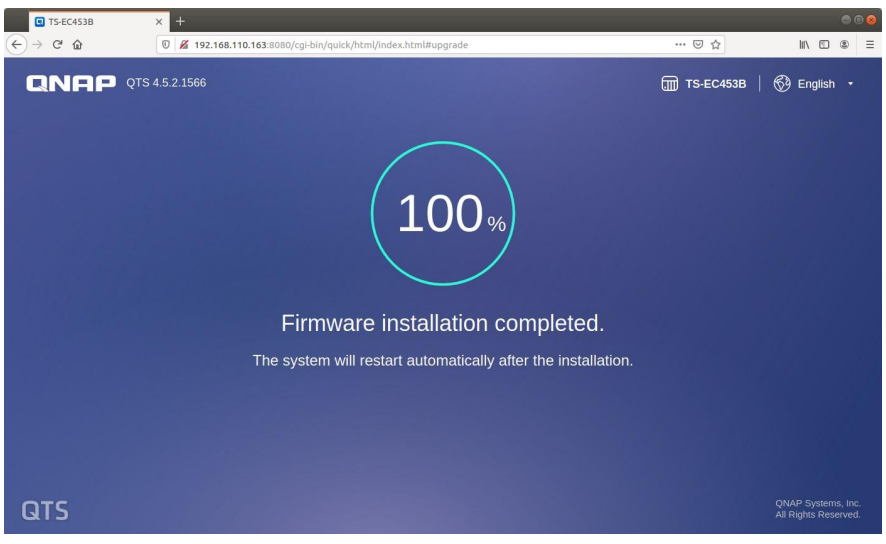
在/home/tc/patch/目录下存放着所有补丁需要的文件。你也可以添加各种自己的补丁文件。【牛年大礼包】工具盘可以把这些补丁文件在启动过程中加载到威廉通的固件系统中。 /home/tc/patch/ 目录中的laojifuli_patch文件会在启动过程中自动执行。可以用于加载各种驱动，例如：网卡驱动，温控驱动等等。也可以用于执行某些特定的补丁程序等等。总之，你可以通过修改laojifuli_patch文件来实现你的要求。这个文件就是留给你的一个特殊的补丁接口文件。为方便你的编程需求而设计的。/home/tc/patch/底下的目录名与威廉通的固件的根目录下的目录名相对应。如果你需要用自己的补丁文件覆盖威廉通固件的文件。可以直接建立对应的目录和文件。启动时即可实现自动加载覆盖。

对于威联通最新版的4.5.2.1566，我使用【牛年大礼包】工具盘进行了测试。可以成功地创建启动盘。测试中发现，该版本有一个非常好新功能。在没有数据盘的情况下，可以自由地升级和降级系统固件。请看如下截图：



通过“Uploaded version”选择固件，就可以直接进行固件升降级。





f541883216 发表于 2021-2-13 08:34 5楼

结束语

本帖最后由 f541883216 于 2021-2-13 09:24 编辑

结束语

很高兴能与大家分享此次的成果。本教程中详细讲解了制作过程中各种细节技术详情。

此次的【牛年大礼包】工具盘，将破解威联通系统提高到了一个新的高度。目前测试了4.5.2.1566最新固安装成功。不但实现了破解，在线固件升级，还实现了序列号等。简直就是可以为任何硬件制作一个“白威联通”系统。不过老骥伏枥再次提醒大家：如果你喜欢威联通系统，请购买正版的“白威联通”系统。不要使用教学用的破解版威联通系统。

文章的最后，老骥伏枥要再次郑重声明：撰写本文与制作【牛年大礼包】工具盘的目的是为了探索，研究，学习嵌入式linux逆向工程技术。禁止利用本文提供的技术和使用本教程的工具盘从事任何非法商业牟利性活动。

参考文献:

[1]【老骥伏枥-狗年大礼包】嵌入式linux逆向工程，手把手教你作黑威联通
[2]【老骥伏枥-原创】制作黑威联通启动盘:进阶篇
[3] Tiny Core Linux -维基百科，自由的百科全书
[4] Super I/O - 维基百科，自由的百科全书

此文绝对是本人【原创】。如果你对本文的观点有不同意见，可以回帖喷我，本人从不固步自封，固持己见。