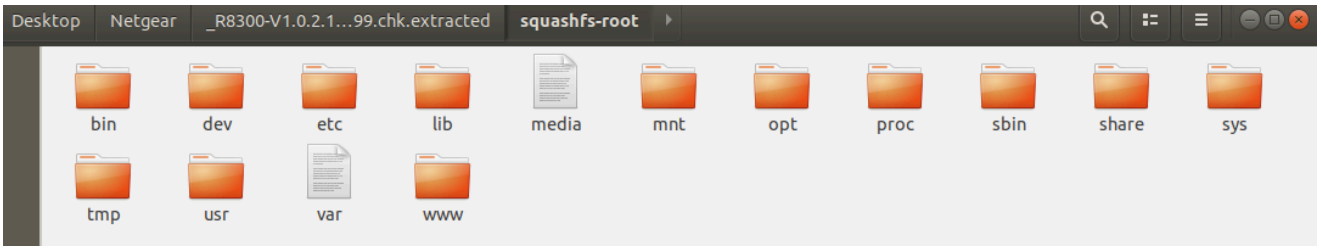
固件下载地址:[夜鹰 X8 R8300 |AC5000 智能 WIFI 路由器 |NETGEAR 支持 --- NIGHTHAWK X8 R8300 | AC5000 SMART WIFI ROUTER | NETGEAR SUPPORT]

基础信息分析:

```
binwalk -Me R8300-V1.0.2.130_1.0.99.chk
```

binwalk解压固件，发现其未加密得到完整的文件系统




```
sudo ./firmwalker-pro-max.sh '/home/iot/Desktop/Netgear/R8300-
V1.0.2.130extracted/squashfs-root' > '/home/iot/Desktop/Netgear/R8300-
V1.0.2.130extracted/firmwalker.txt'
```

firmwalker-pro-max跑一下，看看是否能获得一些有用或可疑信息

```
-------------------- admin --------------------
/etc/avahi-dbus.conf:  <!-- Allow everything, incl
/etc/avahi-dbus.conf:  <!-- Only root or user admi
/etc/avahi-dbus.conf:  <policy group="admin">
/etc/avahi-dbus.conf:  <policy user="admin">
/etc/forked-daapd.conf: admin_password = "unused"
/etc/forked-daapd.conf: uid = "admin"
/etc/group:admin::0:
/etc/system.conf:  <user>admin</user>
```

```
-------------------- root --
/etc/avahi-dbus.conf:  <!--
/etc/group:root::0:0:
```

```
-------------------- password --------------------
/etc/forked-daapd.conf: # Admin password for the non-existent web interface
/etc/forked-daapd.conf: admin_password = "unused"
/etc/forked-daapd.conf: # AirTunes password
/etc/forked-daapd.conf:#        password = ""
/etc/forked-daapd.conf:#        password = "s1kr3t"
```

```
***Search for web servers***
######################################## search for web servers
######################################## httpd
/usr/sbin/httpd


***Search for important binaries***
######################################## important binaries
######################################## tftp
/usr/bin/tftp

######################################## busybox
/bin/busybox

######################################## telnet
/usr/bin/telnet

######################################## telnetd
/usr/sbin/telnetd

######################################## openssl
/usr/local/sbin/openssl

######################################## upnpd
/usr/sbin/upnpd
```

```
file busybox
```

以最典型的busybox为例查看其为32位小端序的ARM架构

```
iot@research:~/Desktop/Netgear/R8300-V1.0.2.130extracted/squashfs-root/bin$ file busybox
busybox: ELF 32-bit LSB executable, ARM, EABI5 version 1 (SYSV), dynamically linked, interpreter /lib/ld-uClibc.so.0, stripped
```

```
checksec --file=busybox
```

发现其除了NX保护(防止在栈和堆等区域执行代码)并没有其它保护

```
iot@research:~/Desktop/Netgear/R8300-V1.0.2.130extracted/squashfs-root/bin$ checksec --file=busybox
[!] Could not populate PLT: invalid syntax (unicorn.py, line 110)
[*] '/home/iot/Desktop/Netgear/R8300-V1.0.2.130extracted/squashfs-root/bin/busybox'
    Arch:     arm-32-little
    RELRO:    No RELRO
    Stack:    No canary found
    NX:       NX enabled
    PIE:      No PIE (0x8000)
```

模拟运行固件:

这里采用qemu系统级模拟
创建虚拟网卡tap0

```
sudo tunctl -t tap0 -u `whoami`
sudo ifconfig tap0 10.10.10.1/24 up
```

```
tap0: flags=4099<UP,BROADCAST,MULTICAST>  mtu 1500
        inet 10.10.10.1  netmask 255.255.255.0  broadcast 10.10.10.255
        ether 6e:73:13:82:f3:32  txqueuelen 1000  (Ethernet)
        RX packets 0  bytes 0 (0.0 B)
        RX errors 0  dropped 0  overruns 0  frame 0
        TX packets 0  bytes 0 (0.0 B)
        TX errors 0  dropped 0 overruns 0  carrier 0  collisions 0
```

启动qemu虚拟机

```
sudo qemu-system-arm -M vexpress-a9 -kernel vmlinuz-3.2.0-4-vexpress -initrd
initrd.img-3.2.0-4-vexpress -drive
if=sd,file=debian_wheezy_armhf_standard.qcow2 -append "root=/dev/mmcblk0p2
console=ttyAMA0" -net nic -net tap,ifname=tap0,script=no,downscript=no -
nographic
```

```
Debian GNU/Linux 7 debian-armhf ttyAMA0

debian-armhf login: root
Password:
Last login: Fri Aug 23 03:24:22 UTC 2024 on ttyAMA0
Linux debian-armhf 3.2.0-4-vexpress #1 SMP Debian 3.2.51-1 armv7l

The programs included with the Debian GNU/Linux system are free software;
the exact distribution terms for each program are described in the
individual files in /usr/share/doc/*/copyright.

Debian GNU/Linux comes with ABSOLUTELY NO WARRANTY, to the extent
permitted by applicable law.
root@debian-armhf:~#
```

给qemu虚拟机配置虚拟网卡，让其与宿主机互通

```
ifconfig eth0 10.10.10.2/24 up
```

```
root@debian-armhf:~# ifconfig eth0 10.10.10.2/24 up
root@debian-armhf:~# ifconfig
eth0      Link encap:Ethernet  HWaddr 52:54:00:12:34:56
          inet addr:10.10.10.2  Bcast:10.10.10.255  Mask:255.255.255.0
          inet6 addr: fe80::5054:ff:fe12:3456/64 Scope:Link
          UP BROADCAST RUNNING MULTICAST  MTU:1500  Metric:1
          RX packets:0 errors:0 dropped:0 overruns:0 frame:0
          TX packets:16 errors:0 dropped:0 overruns:0 carrier:0
          collisions:0 txqueuelen:1000
          RX bytes:0 (0.0 B)  TX bytes:3384 (3.3 KiB)
```

通过wget把打包好的文件系统传入到qemu虚拟机中，并在qemu虚拟机中解压出来，进入到文件系统sq目录中，开始挂载并切换根目录

```
wget http://10.10.10.1:8000/sq.tar

tar -zxvf sq.tar

mount --bind /proc /proc
mount --bind /dev /dev
mount --bind /sys /sys

chroot . sh
```

```
root@debian-armhf:~/sq# mount --bind /proc /proc
root@debian-armhf:~/sq# mount --bind /dev /dev
root@debian-armhf:~/sq# mount --bind /sys /sys
root@debian-armhf:~/sq# chroot . sh


BusyBox v1.7.2 (2018-12-13 12:34:27 CST) built-in shell (ash)
Enter 'help' for a list of built-in commands.

# ls
bin    etc    media  opt    sbin   sys    usr    www
dev    lib    mnt    proc   share  tmp    var
#
```

启动upnp服务发现没有报错服务也没有启动成功

```
# upnpd
#  ./usr/sbin/upnpd
# ps | grep "upnp"
#
```

去分析一下upnpd这个二进制文件，发现需要打开一个upnpd.pid的文件，而我们这里没有，那么我们去创建一个，而var是挂载在tmp/var的

```
v7 = fopen("/var/run/upnpd.pid", "wb+");
if ( v7 )
{
  v8 = getpid();
  fprintf(v7, "%d", v8);
  fclose(v7);
```

```
lrwxrwxrwx  1 iot iot     7 12月 13  2018 var -> tmp/var
drwxr-xr-x  8 iot iot 28672 12月 13  2018 www/
```

所以我们去创建一个tmp/var/run的目录

```
mkdir -p tmp/var/run
```

再次启动upnpd，发现有报错，缺少了nvram，NVRAM(非易失性 RAM) 用于存储路由器的配置信息，而 upnpd 运行时需要用到其中部分配置信息。

```
# /dev/nvram: No such file or directory
/dev/nvram: No such file or directory
/dev/nvram: No such file or directory
/dev/nvram: No such file or directory
/dev/nvram: No such file or directory
/dev/nvram: No such file or directory
/dev/nvram: No such file or directory
/dev/nvram: No such file or directory
/dev/nvram: No such file or directory
/dev/nvram: No such file or directory
/dev/nvram: No such file or directory
```

所以我们需要hook劫持其nvram的动态链接库
网上有现成的库，编译好即可:raw.githubusercontent.com/therealsaumil/custom_nvram/master/custom_nvram_r6250.c
之前file的时候也可以看到采用的是armv5的版本，而我们虚拟机为armv7的版本固还需要用到docker来编译
操作方法:
rootkiter/cross-cpu-compile: 嵌入式 GCC 交叉编译镜像，当前大部分编译器是基于 uclibc 的。产品已经上传至 docker-hub ，可自行参考 README 的相关描述使用。 (github.com)

```
git clone https://github.com/rootkiter/cross-cpu-compile.git
cd cross-cpu-compile/
docker build -t cross-cpu-compile .
docker images
#在docker下执行
```

```
[root@container] # /root/compile_bins/cross-compiler-armv5l/bin/armv5l-gcc -
Wall -fPIC -shared custom_nvram_r6250.c -o nvram2.so
```

加载nvram2.so后发现没有dlsym的符号

```
# LD_PRELOAD="/nvram2.so" /usr/sbin/upnpd
# /usr/sbin/upnpd: can't resolve symbol 'dlsym'
```

发现在libdl.so.0中有dlsym

```
iot@research:~/Desktop/Netgear/R8300-V1.0.2.130extracted/sq$ grep -r "dlsym"
Binary file usr/local/sbin/openvpn matches
Binary file usr/sbin/tc matches
Binary file usr/sbin/afpd matches
Binary file usr/lib/libsqlite3.so matches
Binary file usr/lib/libasound.so matches
Binary file sbin/pppd matches
Binary file lib/libdl.so.0 matches
Binary file lib/libhcrypto-samba4.so.5 matches
Binary file lib/libsqlite3.so.0 matches
Binary file lib/libldb.so.1 matches
Binary file lib/libsamba-modules-samba4.so matches
Binary file lib/libcrypto.so.1.0.0 matches
Binary file lib/libkrb5-samba4.so.26 matches
```

那么我们加载nvram2.so的同时也加载libdl.so.0

```
LD_PRELOAD="/nvram2.so /libdl.so.0" /usr/sbin/upnpd
```

发现又缺少了nvram.ini文件

```
# LD_PRELOAD="/nvram2.so /libdl.so.0" /usr/sbin/upnpd
# [0x00026460] fopen('/var/run/upnpd.pid', 'wb+') = 0x00912008
[0x0002648c] custom_nvram initialised
[0x76f23ecc] fopen('/tmp/nvram.ini', 'r') = 0x00000000
Cannot open /tmp/nvram.ini
```

写一个nvram.ini

```
upnpd_debug_level=9
lan_ipaddr=10.10.10.1
hwver=R8500
friendly_name=R8300
upnp_enable=1
upnp_turn_on=1
upnp_advert_period=30
upnp_advert_ttl=4
upnp_portmap_entry=1
upnp_duration=3600
upnp_DHCPServerConfigurable=1
wps_is_upnp=0
```

```
upnp_sa_uuid=00000000000000000000
lan_hwaddr=AA:BB:CC:DD:EE:FF
```

upnp服务启动成功