Digital Forensics in IoT Security: Detecting Covert Data Exfiltration in Raspberry Pi

Technical Report

Fontys ICT 2024-2025
Cybersecurity Research Group
Mentors: Dr. Mark Madsen, Casper Schellekens
CreateLab Technical Support & Advice: Edwin van den Oetelaar, Jan Dobbelsteen
Student: Artur Kraskov

Contents

Abstract

This research explores covert communication channels in air-gapped systems, inspired by the RAMBO attack from Ben-Gurion University's Cybersecurity Research Lab. This novel technique transforms a CPU into a transmitter, enabling data exfiltration across an air gap. The study aims to detect and mitigate such attacks by leveraging spectrum analysis to monitor and analyze electromagnetic emissions. The ultimate goal is to develop a monitoring system capable of detecting covert channels and implementing countermeasures such as noise interference or active signal cancellation.

Research Problem and Objectives

As cyber-attacks increasingly bypass traditional security mechanisms, air-gapped systems face growing threats from covert data exfiltration via electromagnetic emissions. This research focuses on detecting such attacks, specifically on air-gapped devices like the Raspberry Pi 4. The objective is to develop a comprehensive monitoring and mitigation system for deployment in high-security environments such as financial institutions, research labs, healthcare facilities, law enforcement, and military installations. Enhancing detection and prevention capabilities in these settings strengthens cybersecurity and safeguards critical infrastructure.

Research process description

Stage 0: previous cybersecurity research, nmap port scanning, threat analysis, surveillance, mesh communication and jammers

We began with nmap port scanning [1], guided by reports on internet colonization and large-scale scans by agencies like the NSA [2]. Research then covered surveillance threats in everyday device usage [3], ranging from behavioral manipulation to real-time PsyOps simulations. Jacob Appelbaum’s work [4] and the Hacking Google series [5] further fueled this interest. Brief exploration of mesh communication and device-emitted radio frequencies [6], as well as cybersecurity and privacy-oriented platforms [7], followed.

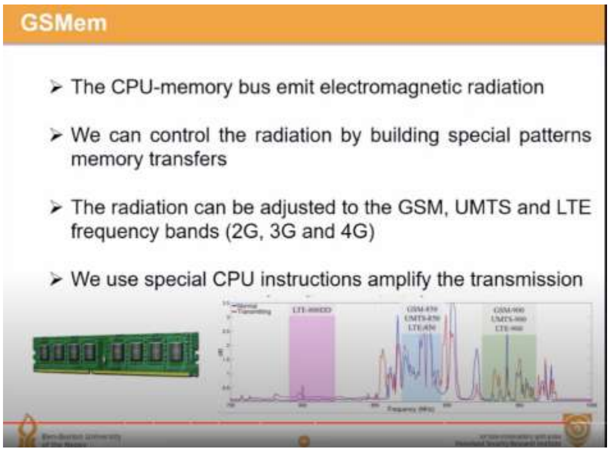
Among possible solutions was a FPGA phone kit [8], promising hardware transparency but still susceptible to persistent attacks. Recognizing device emissions as a surveillance vector, the focus shifted to SDR, hardware emissions monitoring, and eventually air-gapping and covert channel studies [9].

Stage 1: preliminary research, air-gapping & covert channels

The initial research on covert channels, air-gapping, and related attacks originates from the Ben-Gurion Cybersecurity Research Lab [10], a pioneering institution in this field. A specific

YouTube video, *ODINI: Escaping Data from Faraday-Caged Air-Gapped Computers* [11], sparked interest in the topic. Further relevant research was also identified [12].

Additional intriguing videos and concepts were explored on the channel. Ultimately, a comprehensive review of the topic was found in a Black Hat conference presentation by [Dr. Mordechai Guri]() [13]. Receiving computer near the faraday cage room [11].



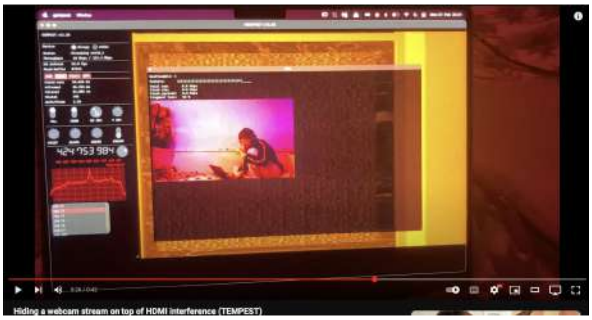
Screenshot explaining CPU emissions of electromagnetic fields in the air-gap context, The Air-Gap Jumpers (18:08) [13]

Dr. Guri emerged as a pioneering researcher in the field. Various videos and papers were found on his website, [www.covertchannels.com]() [14], showcasing a range of air-gapping techniques and methods.

Phone with headphones without internet used to steal data [15]

One of the latest papers at covertchannels.com referred to an air-gapping method utilising screens to leak data with pixels. A computer screen creates pixel patterns captured by another computer with a closed camera [16].

Another screen-based example was identified in connection with TEMPEST and Van Eck phreaking [17].



“A program creates specially crafted artifacts on screen that cause the HDMI cable to radiate MJPEG compressed webcam video.” [17-18]

In summary, covert channels have been demonstrated across various hardware components, highlighting the feasibility of data exfiltration through diverse means. Identified air-gaps and covert channels include:

- **Electromagnetic & Radiation-Based**: Digital noise, EMF, CPU/GPU/RAM emissions, power supply fluctuations
- **Optical**: Light indicators, screens, LEDs, light bulb fields
- **Acoustic**: Speakers, gyroscopes
- **Wired Signal Leakage**: SATA, HDMI, Ethernet cables, and other wiring

These covert channels fall into four main categories: **acoustic, optical, thermal, and electromagnetic**.

The wide range of potential covert channels presents a significant challenge—there is no comprehensive system or universal tool for their detection. Effective monitoring requires analyzing multiple sources and frequency ranges, from hardware emissions to subtle environmental signals like light bulb field variations.

Stage 2: Exploring RAMBO attack, formulating research proposal

M. Guri’s paper, *"RAMBO: Leaking Secrets from Air-Gap Computers by Spelling Covert Radio Signals from Computer RAM,"* served as a key reference and starting point [19]. The proposed research focused on the practical exploration of external EM spectrum monitoring, which was highlighted as a defensive countermeasure in the original study.

Table 7. Defensive countermeasures

Solution	Drawbacks
Zone restrictions (red-black separation)	Cost and space limitation
Host intrusion detection systems (user/kernel)	High rates of false positive
External electromagnetic spectrum monitoring	High rates of false positive
Internal RAM operation jamming	Disruption of the RAM functionality and overhead
External radio jamming of RAM frequencies	Radio interference, high cost, and power consumption
Radio reduction/blocking Faraday enclosures	Cost and maintenance

M. Guri, “RAMBO: Leaking Secrets from Air-Gap Computers by Spelling Covert Radio Signals from Computer RAM” table 7 [19]

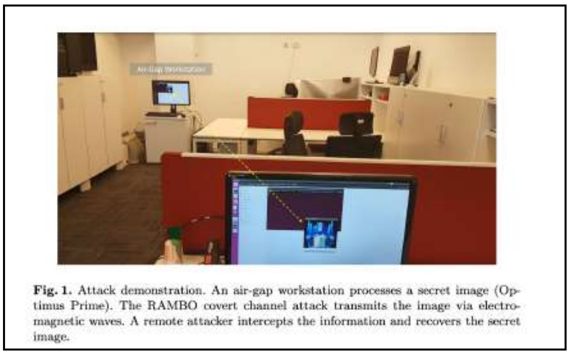


Fig. 1. Attack demonstration. An air-gap workstation processes a secret image (Optimus Prime). The RAMBO covert channel attack transmits the image via electromagnetic waves. A remote attacker intercepts the information and recovers the secret image.

M. Guri, “RAMBO: Leaking Secrets from Air-Gap Computers by Spelling Covert Radio Signals from Computer RAM”, demonstration [19]

Building on prior research and feedback from the Fontys Cybersecurity Research Group, the proposal was refined to focus on hardware emissions and SDR technology [20-21].

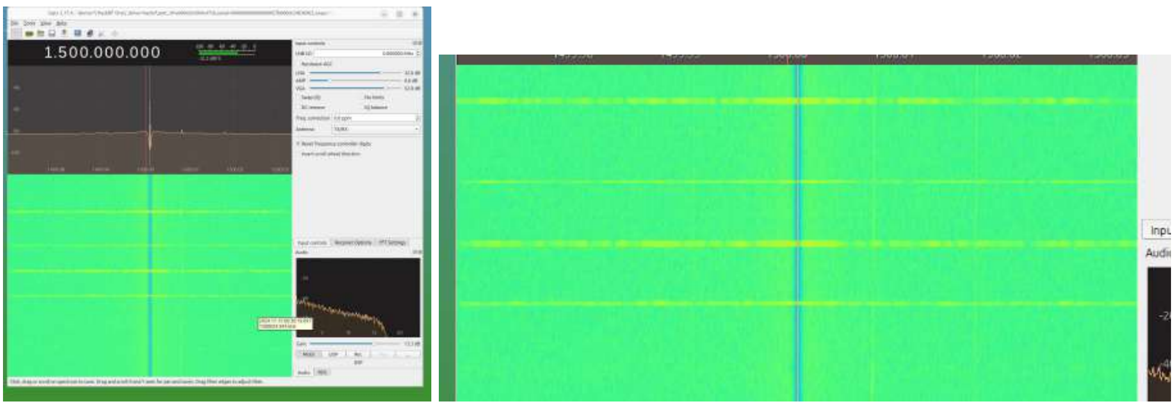
Stage 3: SDR, CreateLab Setup: Raspberry Pi FM transmitter + SDR receiver

HackRF One SDR and an ANT500 (75MHz–1GHz) antenna were borrowed from FHICT ISSD for research [22-23]. Initial tests targeted Wi-Fi and Bluetooth (2.4GHz & 5GHz) using an iPhone SE.

Attempts to capture WhatsApp message transmissions in these frequency ranges were unsuccessful, likely due to the need for advanced tuning. However, connecting or disconnecting the phone from Wi-Fi and Bluetooth produced short bursts of detectable signals in GQRX, notably at 5742000 kHz. These periodic traces indicated emissions during network transitions, though no images were captured.

Using the same setup (HackRF One & ANT500), attempts were made to capture CPU or RAM frequencies under heavy load.

Testing at 1.5GHz—the Raspberry Pi’s CPU frequency as shown in *bpytop*—yielded no detectable CPU or RAM signals. However, following the pattern observed with Wi-Fi and Bluetooth, brief signals were detected during Raspberry Pi reboots.



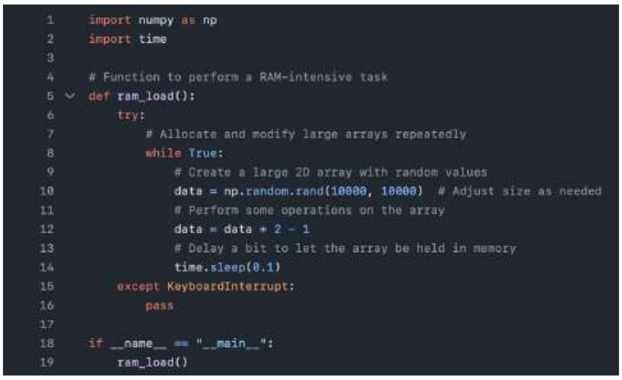
Capturing short signals at 1.5GHz when Raspberry Pi is rebooting

Screenrecord: capturing_1.5GHZ_CPU_reboot.mov [24]

Some signal traces appeared randomly during Pi operation, but the reboot pattern remained consistent across multiple iterations. Testing RAM activity at 500MHz did not yield any detectable signals.

Scripts

CPU heavy load generation script [25]



RAM heavy load generation script [26]

Technical guidance was sought from Fontys CreateLab specialists Edwin van den Oetelaar and Jan Dobbelsteen. As a practical case, the idea of using a Raspberry Pi as an FM transmitter was proposed. A relevant tutorial, including C++ code and a Raspberry Pi 4 pinout, was found online [27-28].

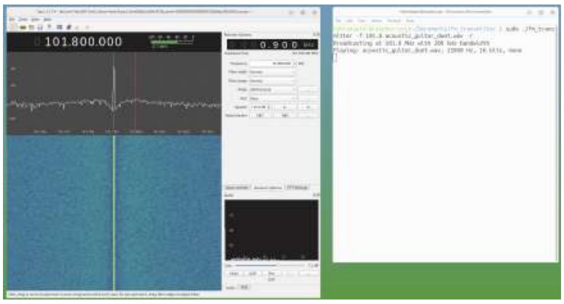
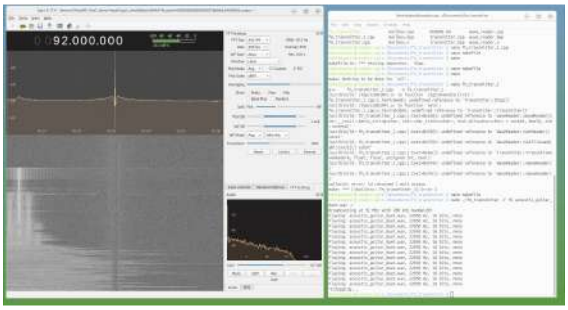
This simplified setup aimed to demonstrate tools and practices for monitoring frequency ranges where covert channels could emerge. Instead of RAM, a GPIO pin with a wire was repurposed as an FM transmitter.

Raspberry Pi with a ground wire on GPIO4 (Top left), HackRF One with antenna (below Pi), Network router used for ethernet communication, project PC driving SDR

HackRF SDR with an ANT500 antenna served as the receiver, with GQRX used for signal monitoring. A script from the internet compiled and ran smoothly, transmitting a melody from a *.wav* file over an adjustable FM frequency.

The setup, located at Fontys CreateLab, was connected via Ethernet and operated remotely using Tailscale (Raspberry Pi) and a VNC server (Project PC).

FM signals from the Raspberry Pi 4 were successfully captured at 92MHz using SDR and GQRX. However, at default frequencies (102, 101.8MHz), no signal was detected, likely due to interference from nearby PCs or overlapping hardware noise.

 Melody is not received with SDR at 101.8MHz, remote shell with Raspberry Pi	 Melody transmission received with SDR at 92MHz

Screenrecord:

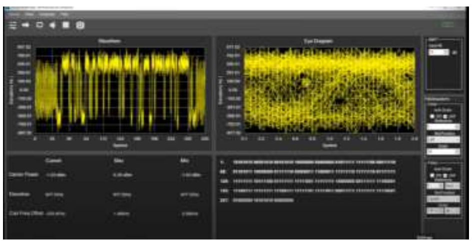
1. capturing_fm_transmission_from_pi_92MHz.mov [29]
2. capturing_fm_transmission_from_pi_92MHz_1.mov [30]

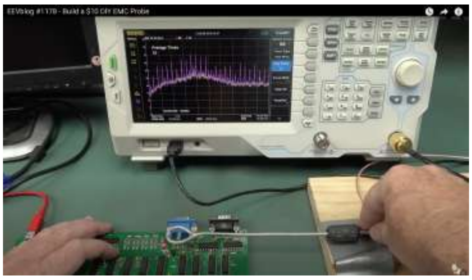
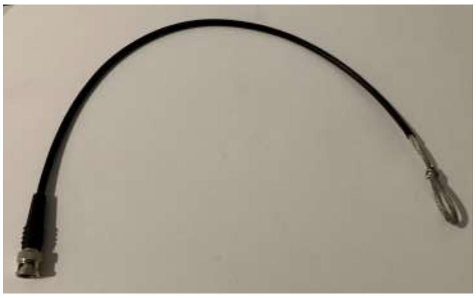
Stage 4: Spectrum Analyzers, near-field probes, amplifiers

Further research highlighted the advanced capabilities of Spectrum Analyzers, including their use in capturing and demodulating Z-Wave signals, commonly found in remote-controlled lamps.

Exploration of near-field probes for EMF and unintentional emissions led to the discovery of a DIY approach using an SMA cable, demonstrated in a video [34]. Following this example, a similar probe was constructed.

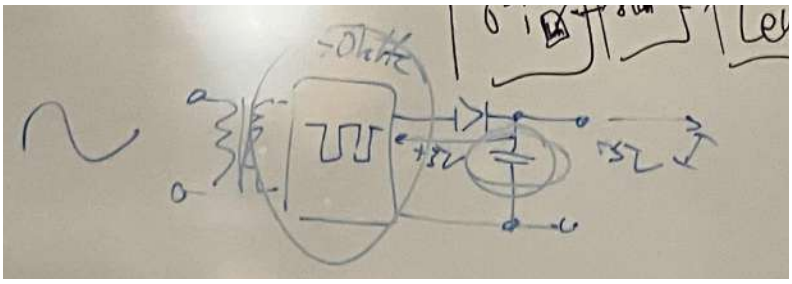
For testing, a **SIGLENT SSA 3032X Spectrum Analyzer** was borrowed from ISSD. However, without an amplifier, the DIY probe failed to capture signals during CPU & RAM load tests and reboots. The video also emphasized the need for an LNA amplifier, which was subsequently found and ordered, along with a set of basic near-field probes.

Setup and reference table		
	Spectrum Analyzer Software demodulating Z-Wave and revealing data [31]	**Reference video:** Using S1220 to De… [31]

	Screenshot from video demonstrating the test of hand made near-field probe [34]	#234: Basics of Ne… [32] Near Field Probe D… [33]
 	DIY near field probe made from SMA cable	EEVblog #1178 - B… [34]
	SIGLENT SSA 3032X Spectrum Analyzer [35]	https://siglentna.com/product/ssa3032x/
	10kHz-6GHz cheap LNA amplifier [36]	R91A Radio Signaalversterker 10K-6GHz Volledige Ranged Low Noise Versterker 9037BAT RF Versterker 600mAh Batterij 5V70ma - AliExpress
	Near-field probes [37]	Aliexpress link

Stage 5: EMC Compliance, power-supply emission & digital noise

Further discussions within the research group and CreateLab led to an intermediate goal: detecting power supply emissions or signals from devices. Since covert channels may exist in devices, understanding their detection could be a crucial first step. This approach aims to distinguish between normal operation and potential compromise.



Simple diagram explaining power supply by Jan Dobbelsteen.

Key questions remained:

- Can power supply emissions be detected remotely?
- How do CPU and component activity reflect in power consumption patterns?

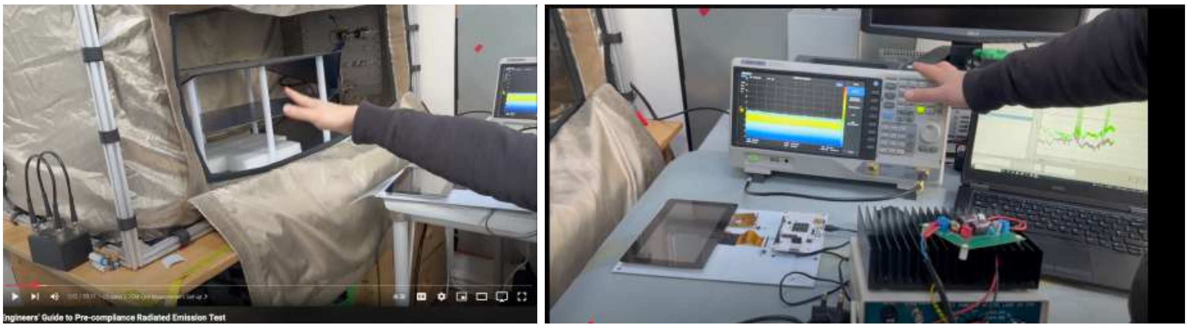
Jan referenced the **EMC Facility** as a relevant research direction, particularly in response to the ODINI attack bypassing a Faraday cage [11]. Unlike the tested setup, EMC facilities use certified Faraday cages designed to be impenetrable.

Consumer electronics and hardware are subject to strict **EMC regulations** to ensure compliance before reaching the market or common environments. Devices undergo **EMC testing and verification** to meet these standards [38].

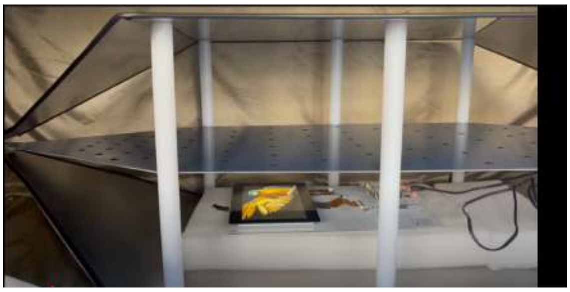
A car inside an EMC facility going through a test [39].

Further EMC research highlighted the use of **directional antennas**, such as the Yagi antenna seen in a referenced image [40], for device localization. A key goal of EMC testing is identifying and mitigating unintentional emissions to ensure devices are secure and interference-free. A relevant video demonstration was also found, providing a step-by-step guide on detecting and measuring power supply emissions during device boot.

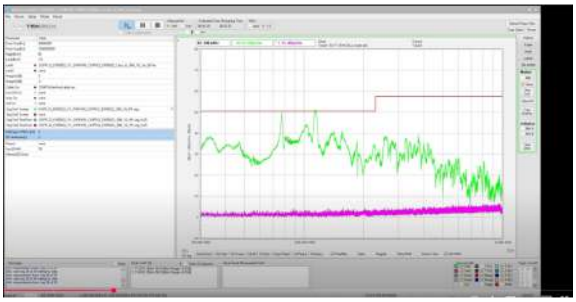
Video: Engineers' Guide to Pre-compliance Radiated Emission Test [41]



Setup demonstration (device, spectrum analyzer, power supply, laptop with software, part of the faraday cage on the left) [41]



Device inside a faraday cage, which must block noise and prevent existing waves from coming in and interfering the measurement [41]



Turn off and turn on sweeps, purple shows spectrum before booting up, green shows device booting up process [41]

A researcher from the video provided additional resources, including a self-study **EMC compliance testing course** [42-45].

This validated the chosen research direction, demonstrating a precise method for detecting **boot emissions** and revealing distinct power-supply patterns across different operations and boot phases. These findings aligned with previous **SDR-based observations (Stage 3)**, where a Raspberry Pi reboot left detectable traces.

A practical verification was conducted using a **Raspberry Pi with an attached power supply**. In the video, the device emitted sound upon boot—an effect previously noted by Edwin, who also shared the recording setup.

Video: raspberry_pi_and_power_supply_boot_sound.MOV [46]

Additional research papers and references on **covert and side channels** provided valuable material for further exploration: Extra reference covert and side channels [47]

Unintentional emissions enable **side-channel attacks**, allowing hardware manipulation via **electromagnetic impulses** or exploiting existing leaks as vulnerabilities. Advanced malware can even detect **near-field probes**, temporarily masking compromised behavior before reverting.

While practical experiments identified potential **covert channel detection** methods, monitoring remained a broad challenge with numerous signals to process. This underscored the need to explore **existing solutions**.

Bug detectors review

A range of **bug detectors and portable spectrum analyzers** were explored, from low-cost options to professional-grade devices.

Entry-level detectors ($6–$10, found on Amazon/AliExpress) include **basic RF, infrared, and EMF detectors**, as well as GPS signal blockers. **Mid-range models** ($16–$160) offer added features like **MAC address identification** and **WiFi scanning**. **High-end solutions**, such as **OSCOR Blue ($33,000)** and **Keysight/REI USA analyzers ($8,000–$42,000)**, require technical expertise and include **log-periodic antennas and AI-powered systems**.

Additionally, **Non-Linear Junction Detectors (NLJD)**, used to locate hidden electronics, were reviewed but lacked listed prices. The study also noted **significant price disparities** among similar devices, suggesting branding and marketing influence pricing beyond functionality.

Comparison Table of Bug Detectors & Spectrum Analyzers

Category	Price Range ($)	Features	Technical Knowledge Required
Entry-Level Bug Detectors	1 - 10	Basic RF/Infrared detection, limited range	None
Mid-Range RF Detectors	16 - 160	Better scanning range, some MAC/WiFi detection	Low
Advanced Spy Detectors	300 - 900	GPS detection, device identification	Medium
Portable Spectrum Analyzers	50 - 5000	Signal analysis, log periodic antennas	Medium to High
High-End Analyzers	8,000 - 42,000	AI-powered, real-time monitoring, advanced UI	High
Professional NLJD	Not Listed	Detects hidden electronics	High

This analysis suggests that users should carefully **evaluate their needs and technical expertise** before investing in a bug detector or spectrum analyzer, as price does not always correlate with effectiveness.

Document with screenshots & links: Bug detectors and portable spectrum analyzers [48]

Bug detectors review conclusion

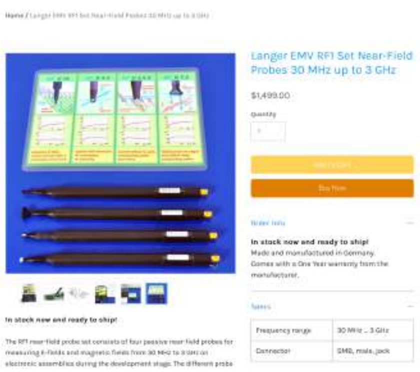
Exploring various options sparked interest in developing **dedicated hardware**, given the **price disparities** and **market demand**. While **high-end equipment** like **Keysight** remains costly, the abundance of **cheaper alternatives** suggests both interest and a lack of awareness about operational principles.

Surveillance concerns drive interest in **privacy protection**, yet no single device ensures comprehensive **covert channel detection**. However, **accurate spectrum analyzers and quality antennas** are more effective in some cases, while **basic portable analyzers with log-periodic or Yagi antennas** suffice for others.

This highlights a future **research direction** in **reverse engineering** and **hardware development** for **EMC compliance, bug detection, and spectrum analysis**. Integrating knowledge from various detection methods could lead to a **more robust covert channel detection and mitigation strategy**.

Stage 6: Low Noise Amplifier, near-field probes, capturing EMF emissions, exploring DIY detectors

Next step was with more practical experiments with probes received from AliExpressand LNA amplifiers. <u>It is worth noting that professional near-field probes can also cost several thousands of euro.</u> Quality allows precision and more clear signal capture, less noise.

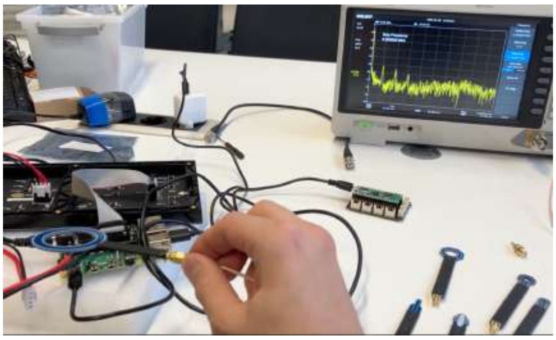


Near-Field probes kit for 1500$
https://www.stratatek.com/product-page/langer-emv-rf1-set-near-field-probes-30-mhz-up-to-3-ghz?srsltid=AfmBOoqHuqeexmv77I4MVolJJnAdRYA9vsOuzPNKliM45h186nldRnEA

Received probes for 10$
Aliexpress link

Obtained a near-field probe kit commonly used for electromagnetic interference (EMI) and electromagnetic compatibility (EMC) testing. Based on the visual inspection of the probes from left to right (image with received probes):

Probe Type	Field Type	Use Case	Frequency Range	Application
Small Loop (1st Probe)	Magnetic (H)	High-resolution detection of currents, loops	10 kHz – few GHz	Locating noise in PCB traces/components
Larger Loop (2nd Probe)	Magnetic (H)	General interference detection over wider areas	Slightly broader than 1st probe	Evaluating coupling, emissions from wider traces
Medium Loop (3rd Probe)	Magnetic (H)	Moderate resolution for larger areas	Lower high-frequency sensitivity	Noise detection where fine resolution isn't needed
Small Widehead (4th Probe)	Electric (E)	Detecting voltage variations, RF interference	MHz – GHz range	Checking voltage noise sources, clock lines, high-speed traces
Large Ring Loop (5th Probe)	Magnetic (H)	Broad area magnetic field detection	Tens of kHz – hundreds of MHz	Emissions from cables, harnesses, large components
Large Circular Loop (6th Probe)	Magnetic (H)	Low-frequency magnetic emissions over large areas	Few kHz – tens of MHz	Compliance testing, emissions from power lines, large enclosures

 Capturing near-field emissions of raspberry pi4 with spectrum analyzer and sweep 0 Hz to 5MHz	Realized Initial test setup at Fontys CreateLab: - Raspberry Pi 4, - Pi Pico, - LED matrix - Near-field probes - LNA Amplifier - Spectrum Analyzer

Video:

1. capturing_near_field_pi_pico_pi_4_with_spectrum_analyzer.mov
2. capturing_near_field_emission_pi_pico_idle.MOV
3. 0Hz_5MHz_sweep_near_field_pico_pi4.MOV
4. near_field_LED_matrix.MOV (in the end touch the matrix with a conductive part, which is incorrect)

These first tests allowed easy capture of existing near-field EMF around Pi Pico, Pi4 and LED RGB matrix. It appeared to be possible to detect near-field emissions of IoT devices in IDLE state, right after they boot.

Stage 7: Sweep with SDR, Python scripting, .csv data collection, exploring DIY EMF detectors

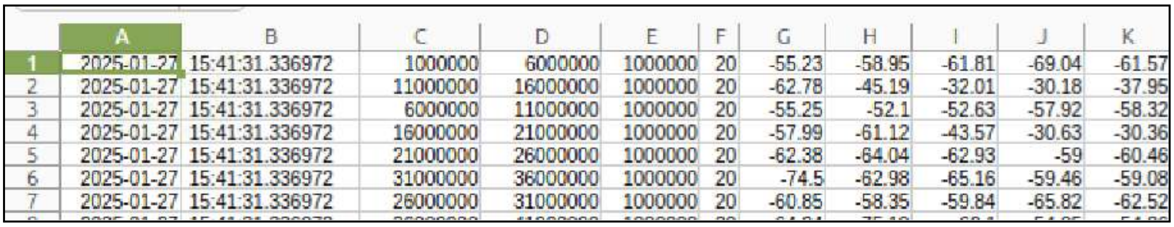
Following successful detection with the **Spectrum Analyzer**, an attempt was made to replicate its capabilities using a more affordable **SDR**.

The setup included **HackRF One**, an **ANT500 (75MHz–1GHz) antenna**, and a **Project PC**. HackRF featured a dedicated CLI tool, **hackrf_sweep**, enabling efficient frequency sweeps and direct data export to **CSV files**.

Bash command: hackrf_sweep -f 75:1000 - to sweep all range from 75MHz to 1000MHz

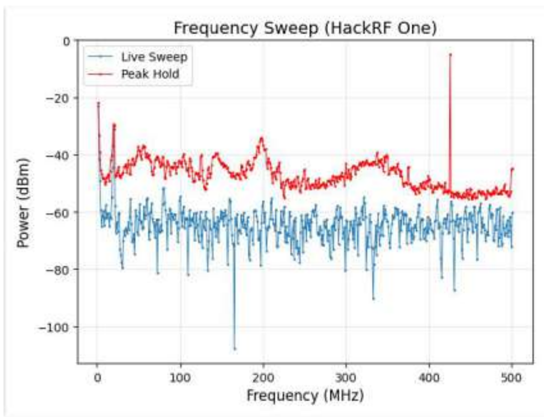
```
2025-01-28, 02:07:45.867619, 990000000, 995000000, 1000000.00, 20, -70.05, -66.51, -61.06, -64.47, -70.54
2025-01-28, 02:07:45.867619, 995000000, 1000000000, 1000000.00, 20, -56.04, -55.78, -59.16, -64.79, -76.50
2025-01-28, 02:07:45.867619, 1005000000, 1010000000, 1000000.00, 20, -64.50, -61.65, -62.87, -67.23, -69.66
2025-01-28, 02:07:45.867619, 1000000000, 1005000000, 1000000.00, 20, -63.33, -63.91, -61.70, -65.98, -66.68
2025-01-28, 02:07:45.867619, 1010000000, 1015000000, 1000000.00, 20, -72.82, -65.41, -56.45, -56.02, -61.97
```

Sweep output in terminal

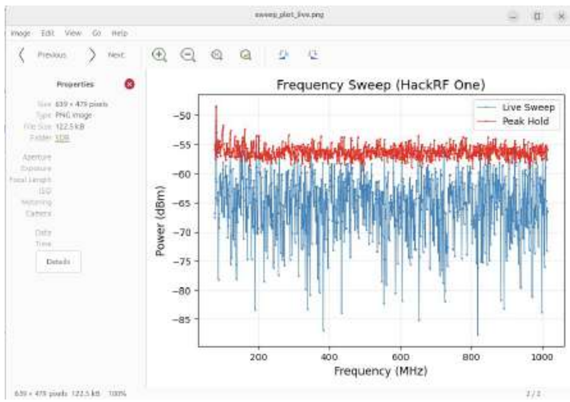


	A	B	C	D	E	F	G	H	I	J	K
1	2025-01-27	15:41:31.336972	1000000	6000000	1000000	20	-55.23	-58.95	-61.81	-69.04	-61.57
2	2025-01-27	15:41:31.336972	11000000	16000000	1000000	20	-62.78	-45.19	-32.01	-30.18	-37.95
3	2025-01-27	15:41:31.336972	6000000	11000000	1000000	20	-55.25	-52.1	-52.63	-57.92	-58.32
4	2025-01-27	15:41:31.336972	16000000	21000000	1000000	20	-57.99	-61.12	-43.57	-30.63	-30.36
5	2025-01-27	15:41:31.336972	21000000	26000000	1000000	20	-62.38	-64.04	-62.93	-59	-60.46
6	2025-01-27	15:41:31.336972	31000000	36000000	1000000	20	-74.5	-62.98	-65.16	-59.46	-59.08
7	2025-01-27	15:41:31.336972	26000000	31000000	1000000	20	-60.85	-58.35	-59.84	-65.82	-62.52

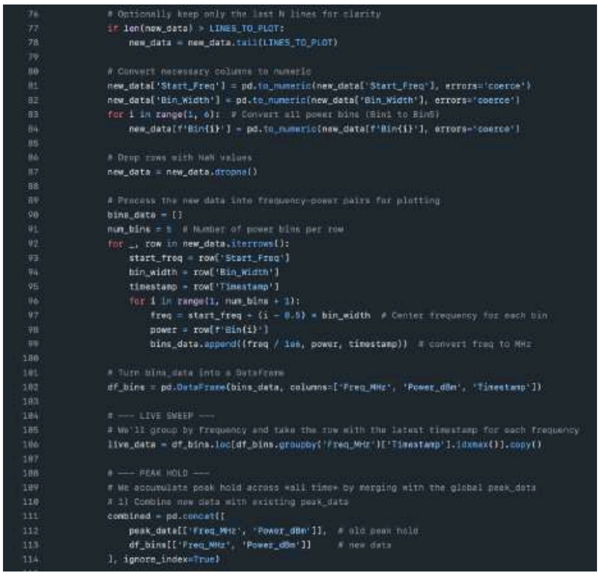
Saved .csv file



Sweep, updatable live image, updates using recent 5-10 sweeps, range is selected with hackrf_sweep 75MHz to 500MHz and recorded



Sweep updatable live image, updates using recent 5-10 sweeps, range is selected with hackrf_sweep 75MHz to 1GHz and recorded



```
# Optionally keep only the last N lines for clarity
if len(new_data) > LINES_TO_PLOT:
    new_data = new_data.tail(LINES_TO_PLOT)

# Convert necessary columns to numeric
new_data['Start_Freq'] = pd.to_numeric(new_data['Start_Freq'], errors='coerce')
new_data['Bin_Width'] = pd.to_numeric(new_data['Bin_Width'], errors='coerce')
for i in range(1, 6):  # Convert all power bins (Bin1 to Bin5)
    new_data[f'Bin{i}'] = pd.to_numeric(new_data[f'Bin{i}'], errors='coerce')

# Drop rows with NaN values
new_data = new_data.dropna()

# Process the new data into frequency-power pairs for plotting
bins_data = []
num_bins = 5  # Number of power bins per row
for _, row in new_data.iterrows():
    start_freq = row['Start_Freq']
    bin_width = row['Bin_Width']
    timestamp = row['Timestamp']
    for i in range(1, num_bins + 1):
        freq = start_freq + (i - 0.5) * bin_width  # Center frequency for each bin
        power = row[f'Bin{i}']
        bins_data.append((freq / 1e6, power, timestamp))  # convert freq to MHz

# Turn bins_data into a DataFrame
df_bins = pd.DataFrame(bins_data, columns=['Freq_MHz', 'Power_dBm', 'Timestamp'])

# --- LIVE SWEEP ---
# We'll group by frequency and take the row with the latest timestamp for each frequency
live_data = df_bins.loc[df_bins.groupby('Freq_MHz')['Timestamp'].idxmax()].copy()

# --- PEAK HOLD ---
# We accumulate peak hold across *all time* by merging with the global peak_data
# 1) Combine new data with existing peak_data
combined = pd.concat([
    peak_data[['Freq_MHz', 'Power_dBm']],  # old peak hold
    df_bins[['Freq_MHz', 'Power_dBm']]     # new data
], ignore_index=True)
```

Python script fragment used for visualization of the data.

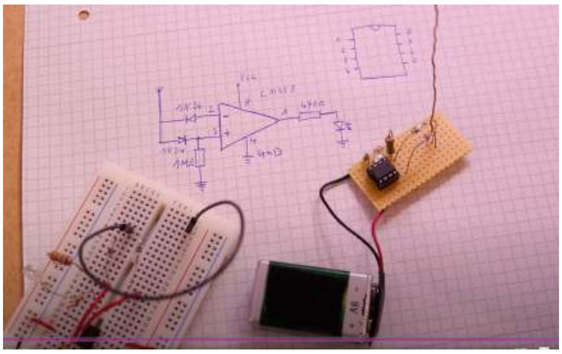
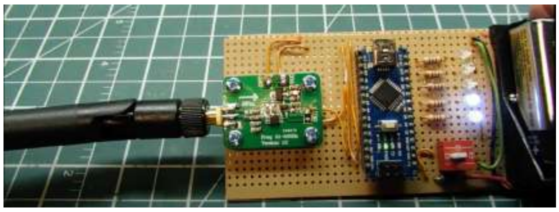
Updated_sweep.py

https://github.com/Krasnomakov/cs_research/tree/main/SDR

Although the available antenna and **hackrf_sweep** script couldn't reliably capture the **Raspberry Pi's FM transmission**, which was previously detected with **GQRX**, this approach still marked progress toward developing a **monitoring system**.

DIY EMF detectors

Next steps encompassed DIY EMF and bug detectors research. And several accessible kits were found. One notable example appeared to be on a ghost hunting channel and used phone pick-up from the radio shack. This special kind of microphone was used with a simple portable voice recorder and a pair of headphones.

 Simple RF detector with LED **Video tutorial:** DIY RF Detector Step by Step build | …	 RF field strength meter **Video tutorial:** Build A RF Field Strength Meter
 EMF Detector circuit DIY Attiny86 EMF [ATtiny85 EMF Detector - Hackster.io] Video tutorial: Attiny85 EMF Detector	 "Homemade EMF Detector For $10!" …

Telephone pick up microphone

- Amazon.com: Telephone Pick-Up Coil with Suction Cup, Features: Record Telephone
- Conversations on Any Tape Recorder with a 3.5 mm Microphone
- Telephone Recording Pickup Coil Suction Cup Microphone 1.5M Cord
- Amazon.com: OM SYSTEM Olympus TP-8 Telephone Pick-up Microphone

As a result, progressing toward the challenge of building a **detection device**, the approach follows a structured process:

1. **Develop and refine hardware components**
2. **Collect data and identify patterns**
3. **Integrate AI and software for advanced analysis**

Step 8: Raspberry Pi EMC Compliance, Near-Field probes and SDR

After consulting **Jan Dobbelsteen** at **CreateLab**, the advice was to **focus on a specific frequency range and attack type**.

This led to a refined approach:

- **Study Raspberry Pi's EMC compliance** and existing standards
- **Analyze individual components** across various operational modes, monitoring **EMF, noise, and emissions**
- **Review IEEE research papers** (accessible via Fontys) for relevant insights.

Two relevant papers were discovered on the internet:
1. Electromagnetic compatibility of Raspberry Pi development platform in near and far-field | IEEE Conference Publication
2. Leveraging Electromagnetic Side-Channel Analysis for the Investigation of IoT Devices | DFRWS

The second paper confirmed that **probes** could be used with **SDR**. Testing validated this, yielding **satisfying results**.

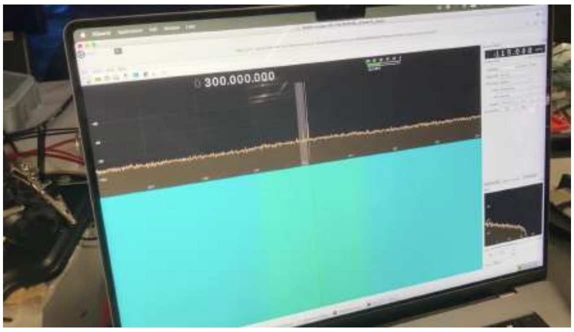
The expected problematic frequencies are following:

- 24MHz (clock out provided to connect additional USB Hubs),
- 25 MHz (crystal for Ethernet connectivity),
- 60 MHz (camera),
- 250 MHz (GPU-Graphic Processing Unit),
- 340 MHz (HDMI-High-Definition Multimedia Interface),
- 450 MHz (SDRAM-Synchronous Dynamic Random Access Memory),
- 900 MHz (ARM-Advanced RISC Machines).

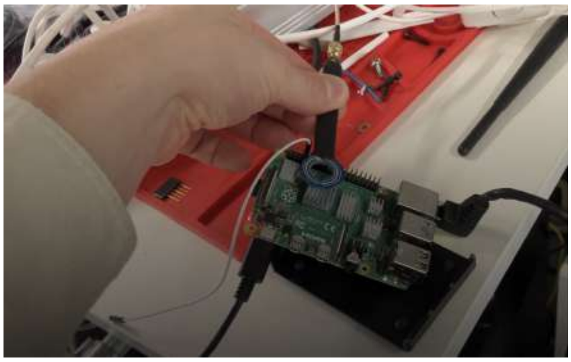
Used theses ranges as a reference from Electromagnetic compatibility of Raspberry Pi development platform in near and far-field | IEEE Conference Publication

The test focused on **300MHz** with a **Raspberry Pi 4**, examining booting, CPU, and RAM under heavy load.

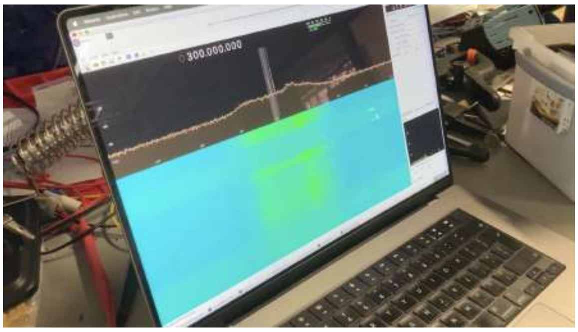
- At **299–301MHz**, the **Pi emitted detectable EMF** upon boot, even in **IDLE**.
- **CPU & RAM load** significantly increased signal visibility.
- **FM transmission from GPIO at 300MHz** was clearly detected.
- **iPhone** showed no response at that time.
- **MacBook Pro** emitted a signal in a specific area while active.



GQRX 300MHz listening with H-field near-field probe, LNA amplifier and HackRF One away from device



Placing the probe near Raspberry PI 4 CPU



Visible signal in GQRX coming with a louder sound

Video:

1. pi4_cpu_heavy_load_emf_ca…
2. raspberry_pi_cpu_300MHz_e…
3. ram_on_heavy_load_emf_cap…
4. pi4_gpio_pin_fm_transmitter_…
5. macBook_pro_300MHz_noise…
6. mac_book_minor_emf_noise_…

Comprehensive Analysis of Signals Intelligence (SIGINT) Monitoring Methods

Researching various types of monitoring eventually led to SIGINT or Signal Intelligence - “the act and field of intelligence-gathering by interception of *signals*” (Wikipedia). A comprehensive summary table was produced.

Category	Key Topics	Techniques & Tools	Challenges & Legal Aspects
System Monitoring & Cybersecurity	Monitoring mobile devices, Wi-Fi, 4G/5G, NFC, Bluetooth	Wireshark, Pi-hole, MDM tools, penetration testing	Limited access due to security restrictions
SDRs (Software-Defined Radios)	Capturing wireless signals (Wi-Fi, ADS-B, LTE)	HackRF, RTL-SDR, GNU Radio	Consumer SDRs struggle with encrypted networks
Electromagnetic Emissions & Covert Channels	Detecting EMI, thermal, and acoustic signals	Near-field probes, thermal cameras, spectrum analyzers	Covert data exfiltration risks

Wi-Fi & Cellular Monitoring	Analyzing 2.4/5 GHz Wi-Fi, 4G/5G signals	Probe request detection, signal analysis	Legal constraints on encrypted transmissions
Practical Cases: Raspberry Pi as Emitter	Generating Wi-Fi and EM emissions	Iperf3, Scapy, stress tests	Hardware profiling limitations
Broader Monitoring Techniques	Radar, radio tomography, UWB sensing	Passive radar, Wi-Fi CSI for motion detection	Requires specialized hardware and software
Advanced Techniques	Side-channel attacks, optical & RF monitoring	TEMPEST, Van Eck phreaking, power analysis	Ethical concerns in surveillance
Legal & Ethical Considerations	Compliance with privacy laws	Monitoring approved networks/devices	Unauthorized interception is illegal
SDR-Based Signal Detection	Configuring SDRs for 2.4 GHz Wi-Fi, LTE	GNU Radio, frequency tuning, signal processing	SDR limitations in high-bandwidth signals
Electromagnetic Profiling	Capturing weak EMI from hardware	Near-field probes, spectrum analyzers	Weak emissions require amplification
Environmental Interactions	Radar-like tracking, Wi-Fi signal interference	Passive radar, signal reflections	Complex environmental noise
Thermal & Optical Monitoring	Infrared heat mapping, screen flicker analysis	IR cameras, brightness sensors	Limited resolution and sensitivity
Acoustic & Ultrasonic Emissions	Detecting device-generated sound waves	High-sensitivity microphones, ultrasonic detectors	Inaudible signals pose analysis challenges
Covert Communication Channels	EM, acoustic, and optical covert data leaks	Spectrum analysis, ultrasonic detection	Requires proximity and specialized tools
Practical Tools & Equipment	SDRs, antennas, spectrum analyzers, near-field probes	GNU Radio, Gqrx, Wireshark, Kismet	High-end tools are expensive

Applications & Use Cases	SIGINT, cybersecurity, device emissions analysis	Passive monitoring, side-channel attacks	Balancing privacy rights with security research
Key Challenges & Limitations	SDR bandwidth, EMI signal weakness, encryption	Signal amplification, noise filtering	Legal barriers in interception
Broader Implications	SIGINT, cyber research, human-device interaction	Smart environments, vulnerability testing	Ethical dilemmas in surveillance technologies

A comprehensive SIGINT and monitoring setup in a lab space involves diverse hardware capable of detecting a wide array of signals and emissions. By understanding the capabilities and applications of each type of equipment, it is possible effectively monitor for:

- **Unauthorized Devices**: Detecting any electronic device attempting to operate within or infiltrate the lab.
- **Human Presence**: Identifying individuals through various detection methods.
- **Environmental Anomalies**: Monitoring changes that could indicate security breaches or equipment malfunctions.

This approach ensures a secure environment by addressing potential threats from multiple angles, leveraging advanced technology to maintain awareness and control over the lab space.

Notes & chat history: All types of SIGINT monitoring [49]

Personal Devices Research: Apple Systems and Security Concerns

System Barriers & Hidden Processes in Apple Devices
Apple's closed design often restricts **low-level access** and **system monitoring**, unlike Linux systems where user control is greater. Despite disabling **Siri**, related background processes persist, raising concerns about **covert processes** or **untraceable backdoors**.

1. **System Integrity Protection (SIP)**
   - Shields key components from user scrutiny.
   - Potentially masks vulnerabilities or hidden processes.
2. **Ongoing Monitoring**
   - During **macOS cybersecurity research**, Siri-related tasks reappeared after being killed by script.
   - **System logs** revealed cryptic **speech sampling** messages with "weight 20.00," indicating a high-priority process invisible to **Activity Monitor**.
3. **Reverse Engineering & Hidden Frameworks**
   - Tools like **Ghidra** can expose hidden components.

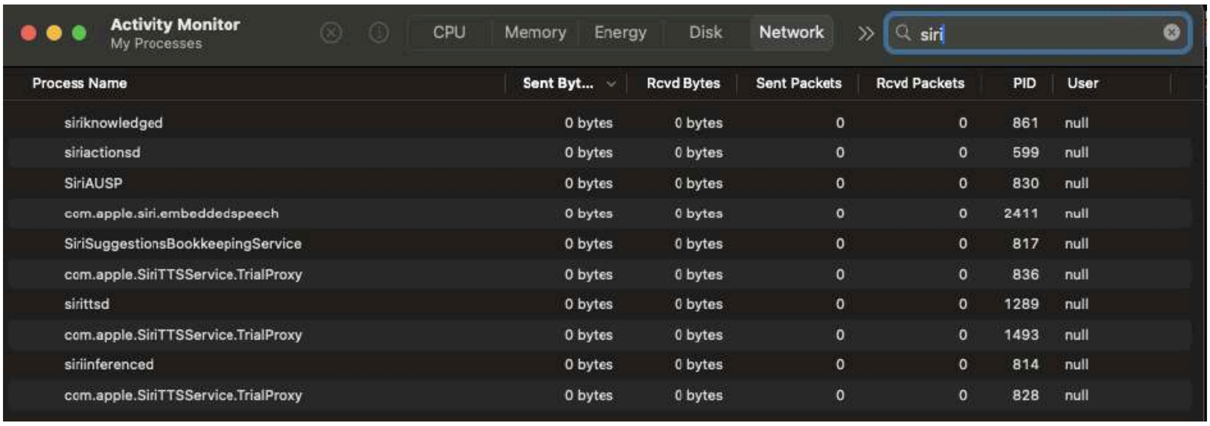
- Example: A **Pegasus** framework (sharing a name with known spyware) remains undeletable under SIP ([Apple Discussion](#)).

4. **Future Directions**
    - Use **reverse engineering** to investigate cryptic processes and potential vulnerabilities.
    - Experiment with **open-source platforms** for maximum control.
    - Continue **system log analysis** to document hidden processes.

By combining personal research with **open-source hardware** and **reverse engineering tools**, it becomes possible to build a **secure, user-controlled** computing environment—critical for both personal privacy and **enterprise security**.

Siri in system logs, when it was disabled in settings

No visible processes after kill siri script (but in system logs still appears)



Siri processes with disabled siri, but without kill_siri script running in the background

Kill_siri script does the work, but potentially can introduce new vulnerabilities. With SIP and protected software it is unclear what is better.

Software & Hardware considerations

Given the variety of **covert channels** and restricted system access, auditing **EMC states** and distinguishing legal operations from **potentially malicious activity** becomes impossible. Hidden **backdoors** or inaccessible vulnerabilities pose security risks. **Open-source, auditable systems** must be prioritized to ensure transparency and control.

Secure Hardware & Network: A Compressed Overview

1. **Hardware from Scratch**
    1. **Build/verify each component** (CPU, RAM, PCB) to minimize **covert channel** risks.
    2. **Production & supply chain security** is challenging but partially feasible via component vetting, controlled processes, and caging.
2. **FPGA or Minimal-Chip Solutions**
    1. Reduce capabilities to lower **side-channel** threats.
    2. Reference: *Research on Cross-board Power-Based FPGA, CPU, and GPU Covert Channels (SpringerLink, iEEE)*.
3. **Open-Source Hardware & Toolchains**
    1. **FPGA-based RISC-V CPUs** (e.g., Litex on GitHub, SiFive HiFive P550).
    2. Avoid proprietary tools (e.g., Vivado, Quartus); **manual flashing** and **emission checks** recommended.
    3. Additional open-source options:
        - BeagleBoard
4. **Router & Network Security**
    1. **OpenWRT** firmware (The Verge Article)
    2. **GL-iNet** routers with optional **GSM modem & VPN** (GL-X750)
    3. Install VPN directly on the router; use **multiple SIMs** for redundancy.
5. **Ready-Made Open-Source Laptops**
    1. Pine64 EU Store / Pine64 Official
    2. MNT Research
    3. Libre Computer
6. **Bug & Spy Detection**
    1. Use advanced RF tools (e.g., **RFeye Node 100-18 (CRFS)**) for **emission monitoring**.
    2. Emission measurements confirm hardware integrity.
7. **Company-Wide Security Approach**
    1. Vet CPU/FPGA supply chain and all infrastructure components.
    2. Combine **open-source hardware**, thorough **verification**, and **monitoring** for robust defense.
8. **Next Steps**
    1. Ensure **secure CPU** or **FPGA** core.
    2. Verify supply chains and **firmware/toolchains**.
    3. Integrate **company-wide policies** covering hardware, software, and operational practices.

By prioritizing **transparent hardware, open-source tools, network security, and comprehensive monitoring**, it's possible to achieve a **secure and functional** system.

Conclusion & Future Directions

Constructing a **comprehensive covert channel detection system** for air-gapped devices remains an ambitious target—approaching **military-level SIGINT** monitoring. Nonetheless, progress in **AI** and **hardware** may eventually shrink the needed resources.

Key Observations

- **EMF Detection**: Devices emit detectable EMF, which shifts under various operations (e.g., boot, CPU load).
- **Network Intelligence**: Some bug detectors can identify **MAC addresses** or device types; known MAC/IP info aids **geolocation**.
- **EMC Profiling**: **Faraday cage** tests enable precise EMF profiling. **Directional antennas** and **near-field probes**can detect devices at greater distances.
- **SDR Versatility**: An **SDR** with a compact antenna can operate like a spectrum analyzer.
- **Practical Challenges**: **Interference** and **ambient noise** complicate real-world EMF detection. Network scans and covert channel checks (heat, light, EMF, sound) often require **multiple steps**.

Active & Preventive Measures

- **Active Cancellation**: Emitters or jammers can disrupt covert channels (e.g., ODINI claims to bypass standard Faraday cages).
- **Secure Hardware**: Crafting **custom devices** (e.g., RISC-V/FPGA) mitigates backdoors and emissions risks.

Research Potential

Balancing **full-spectrum monitoring** against **secure, custom hardware** is a complex, resource-heavy pursuit. Yet, **innovation** is plentiful. Collaboration through **Open Learning** and **AI-driven** tools, along with awareness of threats like **ODINI** and **RAMBO**, can help build robust defenses.

Possible Next Steps

- **Directional Antennas** – Employ **Yagi or log-periodic** antennas for device signal detection.
- **EMC Standards** – Use **EMC testing data** to analyze emissions/compliance for specific devices.
- **Scripting for Device Identification** – Automate **MAC-based** hardware detection and classification.
- **Covert Channel Detection System** – Develop iteratively with **AI**, **applied research**, and **reverse engineering**.
- **Advanced Bug & Spy Device Detection** – Investigate existing tools and **generative AI** applications.
- **Microcontroller & DIY Detectors** – Build and experiment with **chips, antennas, and detection tools** for deeper hardware insight.
- **Signal Differentiation** – Filter noise in varied environments (offices, streets, facilities) and refine detection.
- **Defensive Cybersecurity** – Design secure systems to thwart **air-gap breaches** and covert channels.
- **Hardware Forensics & Attack Reconstruction** – Study real covert channel attacks, reverse-engineer methods, and devise countermeasures.

By combining **low-level hardware research**, **active detection methods**, and **open-source solutions**, a more **transparent, secure**, and **user-controlled** environment becomes achievable.

Reference

1. Kraskov A. (2024). Port scanning with Nmap
2. NSA/GCHQ: The HACIENDA Program for Internet Colonization| Uhr Julian Kirsch, Christian Grothoff, Monika Ermert, Jacob Appelbaum, Laura Poitras, Henrik Moltke https://www.heise.de/hintergrund/NSA-GCHQ-The-HACIENDA-Program-for-Internet-Colonization-2292681.html?view=print&hg=1&hgi=14&hgf=false
3. Kraskov A. (2024). Cybersecurity & Threat Intelligence
4. Appelbaum, J. R. (2022). Communication in a world of pervasive surveillance
5. HACKING GOOGLE Series - YouTube
6. Kraskov A. (2024)., notes Cell phone signal jammers, signal boosting and meshes
7. Kraskov A. (2024)., notes CyberSecurity & Privacy oriented devices
8. FPGA phone kit iCE40 Ultra Mobile Development Platform | Lattice Kits & Boards
9. Kraskov A. (2024)., notes Initial Air-Gapping, Covert Channels and SDR data
10. Ben Gurion Cybersecurity Research Cyber@BGU
11. ODINI: Escaping data from Faraday-caged Air-Gapped computers
12. [1802.02700] ODINI : Escaping Sensitive Data from Faraday-Caged, Air-Gapped Computers via Magnetic Fields
13. The Air-Gap Jumpers (18:08) The Air-Gap Jumpers
14. Research in covert channels www.covertchannels.com
15. How to leak sensitive data from an isolated computer (air-gap) to a near by mo…
16. M. Guri, "PIXHELL Attack: Leaking Sensitive Information from Air-Gap Computers via 'Singing Pixels'" 2024 IEEE 48th Annual Computers, Software, and Applications Conference (COMPSAC)
    Demo video: https://www.instagram.com/reel/C_0Osn6JwHJ/
17. Hiding a webcam stream on top of HDMI interference (TEMPEST)
18. WindyTan's blog post Using HDMI EMI for fast wireless data transfer
19. M. Guri, "RAMBO: Leaking Secrets from Air-Gap Computers by Spelling Covert Radio Signals from Computer RAM" Nordic Conference on Secure IT Systems
    RAMBO = (Radiation of Air-gapped Memory Bus for Offense)
    Demo video: https://youtu.be/BLJcUXd2ny
    Research Coverage video from "low Level Learning:
    https://www.youtube.com/watch?v=ihtAijebU-M
    (http://arxiv.org/abs/2409.02292)
20. Kraskov A. (2024).
21. Kraskov A. (2024). Digital Forensics in IoT Security v2
22. Great Scott Gadgets, HackRF One h Digital Forensics in IoT Security v1
    ttps://greatscottgadgets.com/hackrf/one/
23. ANT500 antenna Telescopic Antenna SMA - 75 MHz to 1 GHz
24. Kraskov A. (2024). capturing_1.5GHZ_CPU_reboot.mov
25. CPU heavy load generation script on personal GitHub:
    https://github.com/Krasnomakov/cs_research/tree/main/cpu_load

26. RAM heavy load generation script on personal GitHub: https://github.com/Krasnomakov/cs_research/blob/main/ram_load/ram_load.py
27. Tutorial: How to Build a Raspberry Pi FM Radio Transmitter
28. Raspberry Pi FM Radio source code on public GitHub: https://github.com/markondej/fm_transmitter/tree/master
29. capturing_fm_transmission_from_pi_92MHz.mov
30. capturing_fm_transmission_from_pi_92MHz_1.mov
31. Using S1220 to Demodulate a Z-Wave Signal
32. #234: Basics of Near Field RF Probes | E-Field & H-Field | How-to use
33. Near Field Probe Demo
34. EEVblog #1178 - Build a $10 DIY EMC Probe
35. SIGLENT SSA 3032X Spectrum Analyzer https://siglentna.com/product/ssa3032x/
36. LNA amplifier R91A Radio Signaalversterker 10K-6GHz Volledige Ranged Low Noise Versterker 9037BAT RF Versterker 600mAh Batterij 5V70ma - AliExpress
37. Near-field probes https://nl.aliexpress.com/item/1005008018625027.html?spm=a2g0o.productlist.main.65.33396813P69KrL&algo_pvid=2113f4f2-4f71-439e-a140-e046f204d80e&algo_exp_id=2113f4f2-4f71-439e-a140-e046f204d80e-32&pdp_npi=4%40dis%21EUR%2134.33%2118.19%21%21%2135.08%2118.59%21%402103835e17380072337993621e0d1d%2112000043268085036%21sea%21NL%210%21ABX&curPageLogUid=UyOj6xS8wZwP&utparam-url=scene%3Asearch%7Cquery_from%3A
38. Electromagnetic compatibility wiki https://en.wikipedia.org/wiki/Electromagnetic_compatibility
39. Car inside an EMC facility image https://performancedrive.com.au/jlr-opens-electrical-interference-test-facility-for-future-vehicles-0617/
40. Yagi antenna wiki https://en.wikipedia.org/wiki/Yagi%E2%80%93Uda_antenna
41. Engineers' Guide to Pre-compliance Radiated Emission Test
42. https://www.youtube.com/@MachOneDesignEMC/videos
43. https://emccompliance.co.uk/training/
44. https://mach1design.co.uk/
45. https://emccompliance.co.uk/
46. raspberry_pi_and_power_supply_boot_sound.MOV
47. Kraskov A. (2024). Extra reference covert and side channels
48. Kraskov A. (2024). Bug detectors and portable spectrum analyzers
49. Kraskov A. (2024). All types of SIGINT monitoring