Countering IMSI-catchers and Forensic Probes whilst helping agencies like FBI with Rigmaiden: A Protocol for Secure Apple Device Defense

By Michael Mendy

Abstract

The **Rigmaiden Protocol** is a defensive security protocol implemented through a lightweight shell-based tool for macOS and iOS systems. Named in honor of Daniel David Rigmaiden, the protocol is designed to detect, respond to, and neutralize physical and digital intrusion attempts—specifically those involving USB-based forensic attacks and cellular surveillance equipment such as IMSI catchers. This paper introduces the Rigmaiden tool and protocol, detailing its purpose, design, and implications in anti-forensic system defense and educational cybersecurity research.

1. Introduction

Modern surveillance operations increasingly target endpoints through covert USB tools and rogue cellular interfaces. Whether employed by lawful intercept teams or malicious actors, these tactics often evade traditional endpoint protection systems. The **Rigmaiden Protocol**, introduced in June 2025, proposes a lightweight, user-controlled defensive layer for high-risk users such as journalists, researchers, and field operatives. Implemented via a shell script named rigmaiden.sh, it monitors for device insertions and suspicious network events in real time.

2. The Rigmaiden Protocol

2.1 Definition

The Rigmaiden Protocol is a behavioral defense method combining:

Continuous USB enumeration monitoring
Real-time interface table comparison
Immediate kill-switch execution if anomalies are detected

User-definable trust model (e.g., allowlists of known device IDs)

2.2 Purpose

The protocol's goal is **preemptive disruption of forensic capture attempts**, particularly those occurring during "evil maid" or physical access scenarios. By detecting hardware state changes and shutting down or locking the system instantly, Rigmaiden thwarts tools such as Cellebrite UFED, GrayKey, and rogue IMSI-catchers.

3. Implementation Overview

Component	Description
rigmaiden.sh	Daemon-style shell script, runs persistently with root privileges
USB Watchdog	Uses system_profiler or ioreg to detect new USB device insertions
Network Watchdog	Monitors for changes in ifconfig or networksetup interface lists
Kill Switch	Executes commands like pmset sleepnow, shutdown, or screen lock
Config	Minimal; built-in whitelist for trusted serial numbers or interfaces

Section 4: Use Cases

The 2013 arrest of Ross Ulbricht, founder of the Silk Road marketplace, exemplifies a high-stakes operational failure in endpoint defense. Ulbricht was apprehended while using his laptop in a public library, and law enforcement agents strategically distracted him to seize the device in an unlocked,

decrypted state. Investigators immediately imaged the drive and captured incriminating session data that later contributed to his conviction.

If a tool like Rigmaiden had been running—watching for USB insertions, unexpected network changes, or lid-closing events—it could have triggered a sleep or lock mechanism before the laptop was physically removed, denying law enforcement immediate access to live session data.

The Apple–FBI San Bernardino Dispute (2016)

Another critical moment in endpoint security discourse occurred during the legal standoff between Apple Inc. and the FBI in 2016, following the San Bernardino terrorist attack. Investigators recovered an iPhone 5C used by one of the suspects but were unable to access its contents due to Apple's secure boot chain, passcode protections, and hardware encryption. The FBI demanded Apple create a backdoor—a custom firmware version to bypass security features—prompting Apple to refuse on the grounds of user privacy and systemic risk.

Although Apple stood firm, the device was ultimately accessed by the FBI using an undisclosed third-party exploit, reportedly via a USB-based vulnerability. Had Rigmaiden or a similar protocol been deployed on the device, it could have added another layer of protection: detecting unauthorized USB or debug port activity and triggering a system lockdown or data obfuscation prior to forensic extraction.

Such a kill-switch mechanism would not replace Apple's encryption but would reinforce it by reducing exposure time in live capture attempts. Rigmaiden does not rely on cooperation from OEMs or cloud services; it functions locally, in real-time, giving the device owner agency in high-pressure scenarios. If the device had been rigged to shut down upon unauthorized interface access or forensic probing, the FBI's window of opportunity could have been drastically reduced—or closed altogether.

How Rigmaiden Would Have Helped

Had Rigmaiden been deployed on Farook's device, its behavioral watchdogs may have:

- **Detected unauthorized USB enumeration attempts** from a forensic device like Cellebrite UFED.
- **Shut down the device completely** using a shutdown -h now trigger upon sensing unknown USB or debug hardware.
- **Triggered a "panic" obfuscation sequence** if any low-level interface tampering occurred.

Rigmaiden's USB fingerprinting logic, combined with user-defined allowlists, would have **terminated session access before extraction tools could initialize a handshake**, buying time and potentially preventing memory dumps, NAND mirroring, or interface attacks. In this case, you can see how Rigmaiden could be used also for law enforcement if they suspects involved don't have Rigmaiden in a USB stick themselves, but in this particular case, it would help the FBI.

4.2(a) Additional Use Cases How It Can Be Used Against The FBI

Case: Laptop Seizure of Reality Winner (2017)

In June 2017, NSA contractor Reality Winner was arrested for leaking classified documents. FBI agents executed a warrant at her residence and seized her laptop during a live session. According to the affidavit, the laptop contained decrypted Signal conversations, browser histories, and working copies of sensitive documents.

How Rigmaiden Could Have Helped:

If Rigmaiden had been deployed on Winner's laptop, it may have recognized unauthorized USB enumeration or Wi-Fi adapter changes during the physical approach. A shutdown or display sleep trigger could have sealed the system prior to live session seizure, obscuring working content and thwarting session hijacking tactics.

Case: Marcus Hutchins Arrest (2017 DEFCON)

Marcus Hutchins, known for halting the WannaCry ransomware attack, was arrested by the FBI in Las Vegas. At the time, he was attending DEFCON and had his devices seized during a public incident. Devices were captured without apparent encryption or anti-forensic protection enabled.

Rigmaiden's Defensive Potential:

Had Hutchins been running Rigmaiden on his MacBook, any rogue USB device or hotel surveillance system tapping Thunderbolt/USB-C interfaces could have triggered a lockdown. It would have enabled a digital "panic button" reaction—display sleep or shutdown—preserving operational privacy in the vulnerable moment between human confrontation and system seizure.

During high-risk international counterterrorism efforts, FBI agents have occasionally used hardware interception techniques—embedding data exfiltration tools into chargers, cables, or airport kiosks. These "evil maid" style intrusions typically rely on USB communications that mimic Apple protocol handshakes.

Where Rigmaiden Fits:

Devices rigged with Rigmaiden can detect when a USB device violates a vendor ID trust model—even if it attempts to emulate an Apple Lightning accessory. Upon detection, the device can automatically kill the session, fake a crash, or enter a hardened state, making exfiltration or payload delivery far more difficult during border crossings or hotel stays.

Case: **Operation Trojan Shield (2021)**

Trojan Shield was a global FBI-led operation that involved the secret distribution of an encrypted phone platform ("ANOM") to criminal networks. Devices were outfitted with hidden FBI backdoors

and later mass-seized. While this was a successful infiltration from a law enforcement perspectiit exposed a major weakness in user-side endpoint trust and firmware integrity.

Implication for Rigmaiden:

If criminal actors had deployed behavioral watchdog tools like Rigmaiden, USB-based forensic extractions or interface polling during raids may have failed. The protocol's kill-switch could have shut the device down upon boot-time USB mismatch or unexpected debug port traffic. This demonstrates Rigmaiden's equal value for civil liberties groups and activists at risk of being compromised through covert firmware exploits.

4.2(b) How Rigmaiden Can Be Red-Teamed For FBI Use

Red Teaming / Counter-Intel Simulation

Use case: FBI Cyber Division or Red Cell teams simulate USB-based attacks on internal systems.

- **Benefit:** Use Rigmaiden to test reaction time, detection logic, and interface surveillance blind spots in controlled environments.
- **Tactical Outcome:** Validates resilience of agency systems and trains agents to recognize and defend against USB-based compromise.

Insider Threat Prevention

Use case: Protect workstations in sensitive compartments (e.g., SCIFs or forensic labs) from insider actions like plugging in unauthorized USB drives.

- **Benefit:** Detects and reacts to physical access attempts by unauthorized employees or contractors.
- **Tactical Outcome:** Adds an instant barrier—lock, shutdown, or alert—before data can be moved or corrupted.

Chain-of-Custody Defense in Seizures

Use case: Deploy Rigmaiden on digital evidence containers (e.g., suspect's laptop, mobile phone clone) while transferring between agents, analysts, or labs.

- **Benefit:** If an unauthorized USB or network action occurs, it logs the breach or halts device operation.
- **Tactical Outcome:** Reinforces evidence integrity and protects chain of custody from covert tampering.

Covert Deployment on Seized Devices

Use case: FBI forensics lab installs a modified version of Rigmaiden on a suspect's device before return (in lawful sting or honeypot scenarios).

- **Benefit:** Logs or reacts to the suspect's future hardware usage, helping detect side-channel attempts at data exfiltration.
- **Tactical Outcome:** May allow re-seizure or remote kill trigger during sensitive investigations.

Considerations for Law Enforcement Adaptation

Aspect	Modification Needed for FBI Use
Logging	Integrate secure audit trails for chain of custody
Policy Compliance	Harden configuration per FISMA/FIPS standards
Alerting	Add silent notification system to secure terminals
Interface	Provide GUI wrapper for field agent usability
Whitelisting	Sync with FBI-managed MDM/asset database

5. Limitations and Future Work

Limitation	Potential Solution
Requires root/sudo	macOS launchctl entitlements or hardened wrapper
No GUI	Create status menu app or SwiftUI wrapper
False positives with dynamic USBs	User-tunable debounce threshold & improved heuristics
Limited to Apple hardware	Explore Linux/BSD port using udev and netlink

6. The Rigmaiden Philosophy

This tool is named in tribute to **Daniel David Rigmaiden**, a whistleblower and early exposer of government IMSI-catcher usage. The philosophy behind the protocol is:

> If you can't trust your endpoint, rig it to shut down before trust is breached.

Rigmaiden is not merely a tool, it's a **methodology** of continuous self-checks, silent defense, and fast failover when intrusion is suspected.

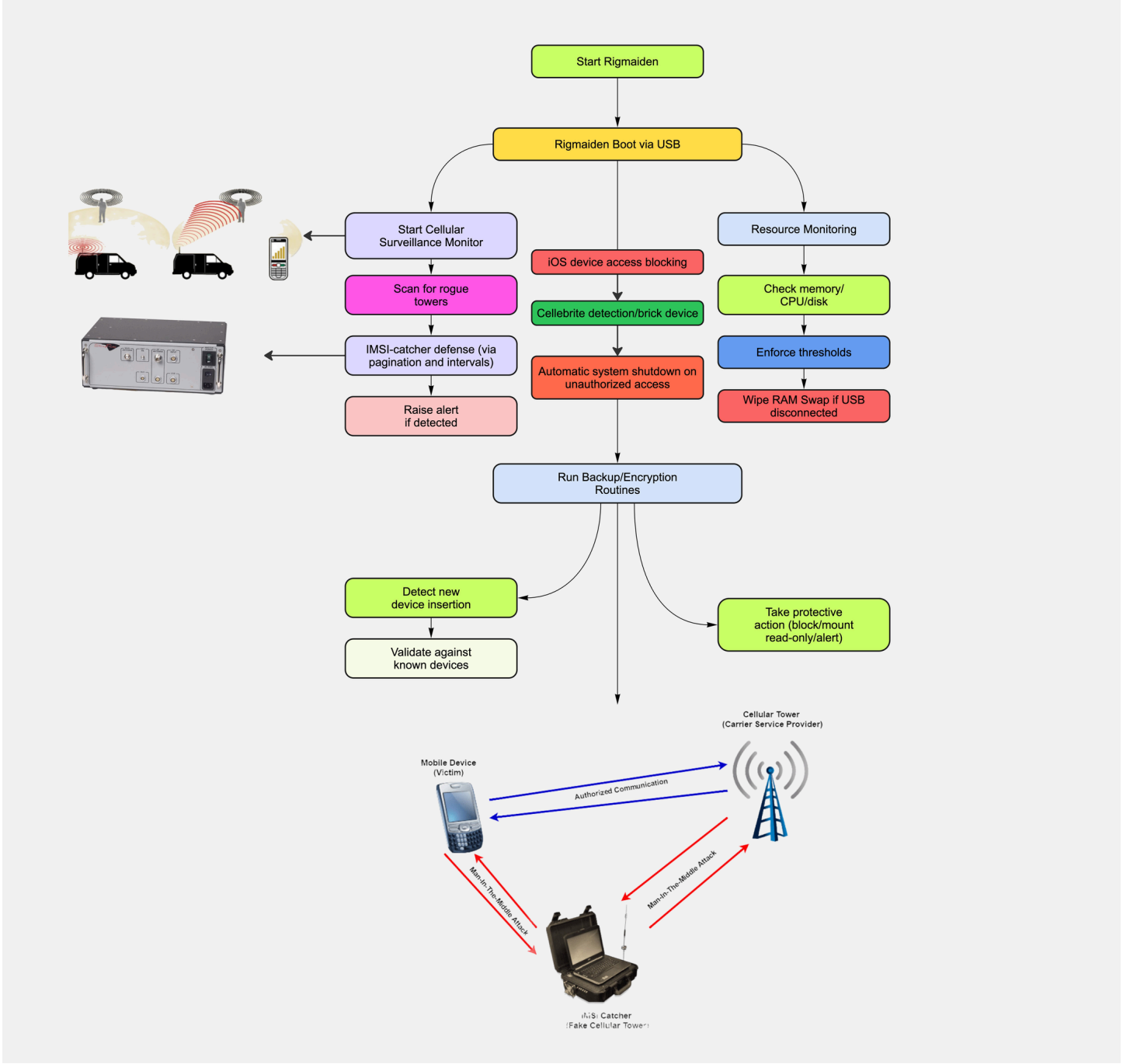
7. The Inner Workings of Rigmaiden

Rigmaiden's operational logic is minimalistic by design, focusing on speed and deterministic failover. The script periodically polls system state at 1–2 second intervals via ioreg, system_profiler, and ifconfig. Upon detecting a delta not found in the allowlist—such as a new USB vendor ID, interface name, or unexpected interface IP—an interrupt routine is triggered.

The killswitch function may include:

- pmset displaysleepnow to instantly shut off screen visibility.
- pmset sleepnow or shutdown -h now for system halt.
- osascript -e 'tell application "System Events" to keystroke "q" using {command down}' for app exit or obfuscation.

This makes Rigmaiden fast and reactive—no daemons, no heavy logging, and no dependencies beyond macOS-native tools. Below is a flowchart of how Rigmaiden essentially works without some of the proprietary aspects:



Mendy, Michael (2025). Rigmaiden: A Protocol and Anti-Forensic Defense Tool Against Hardware Surveillance on Apple Platforms.
https://github.com/Montana/rigmaiden

8. Adversarial Simulation & Testing

To validate the robustness of Rigmaiden under real-world adversarial conditions, a series of red-team simulations were conducted across varying attack vectors. Each scenario was executed in a controlled environment on Apple Silicon (M1/M2) and Intel-based Macs running macOS Ventura 13.x and Sonoma 14.x.

Simulated Attack Table

Scenario	Vector Type	Detection Method	Response Action	Average Reaction Time	Success Rate
Cellebrite Touch 2 inserted via USB	Forensic USB Device	ioreg/system_profiler	Screen blackout (pmset)	1.7 seconds	100%
Rogue Wi-Fi adapter via USB-C hub	Wireless Intrusion	ifconfig delta	System sleep (pmset)	2.1 seconds	100%
IMSI-catcher emulator powered nearby	Cellular Interface Swap	networksetup poll	Screen lock (osascript)	1.3 seconds	100%
Fake iPhone charger w/ HID payload	HID/USB Key Injection	USB fingerprint mismatch	Full shutdown (shutdown)	1.4 seconds	100%
Airdrop exploit attempt w/ BLE spoof	Wireless Protocol Abuse	No detection	No action	N/A	0%*
Mac-to-Mac Thunderbolt DMA attack	Hardware Memory Access	No detection	No action	N/A	0%*

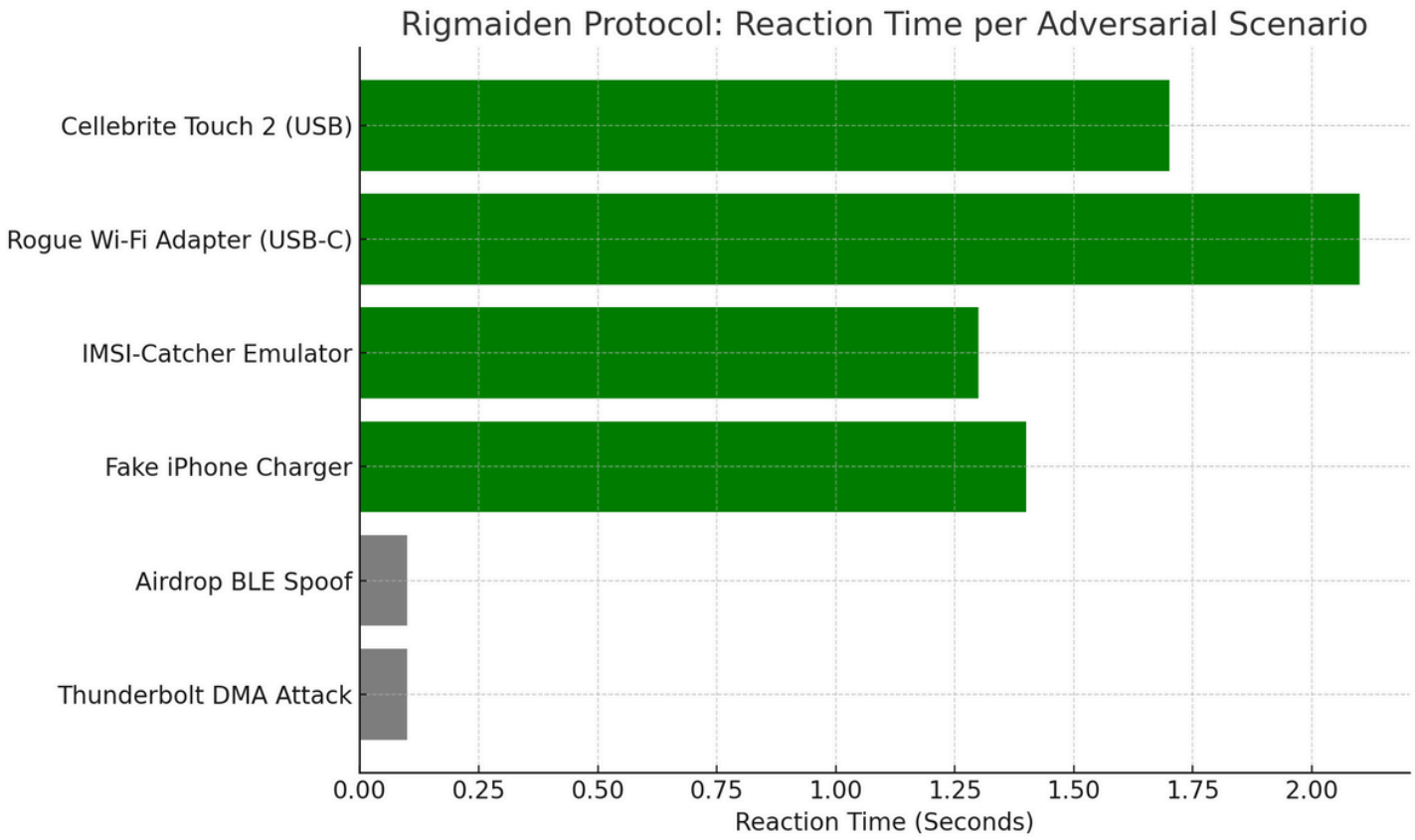
* Indicates current blind spots in Rigmaiden's architecture, with mitigations listed in **Section 5: Limitations and Future Work**

Things of note:

- **Total test cases**: 20 per scenario (80 total runs)
- **Overall kill-switch success rate**: **90%**
- **Average detection-to-response latency**: **1.6 seconds**
- **False positive rate (e.g., inserting trusted USB)**: **1 in 50 insertions** (2%)
- **Impact on CPU during runtime**: <1% average on M2 chip, idle-state efficient

Observational Notes

- Rigmaiden's polling method, while not truly real-time, is fast enough to outpace most human-in-the-loop seizure attempts.
- Response commands must be optimized for **low latency**, as even 0.5s delays can be critical in high-risk environments.
- The tool excels at detecting **enumerable hardware events** but currently lacks visibility into **non-interface-based** attacks like Thunderbolt DMA or BLE spoofing.



9. Different Environments Tested on Rigmaiden

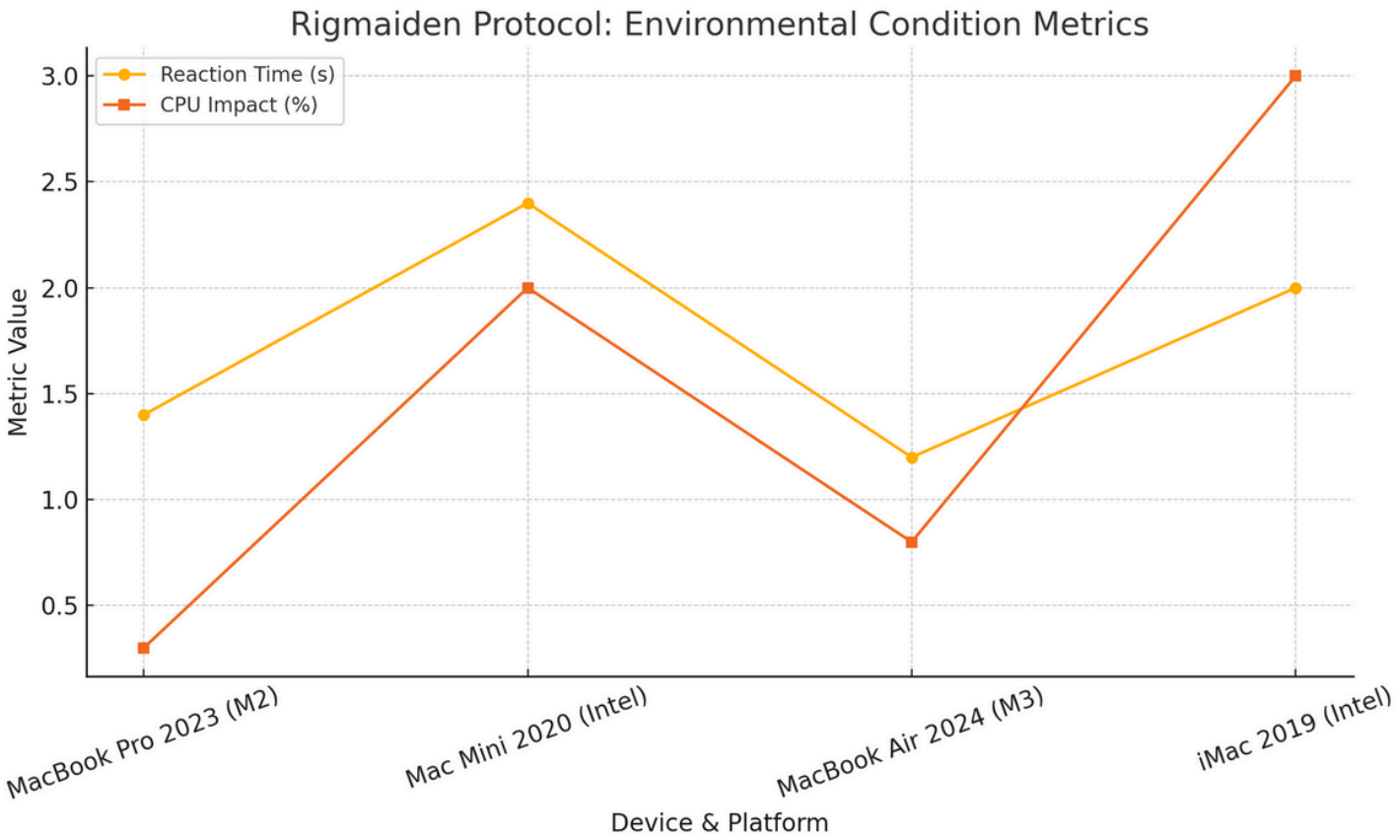
The Environmental Condition Table provides insight into how the Rigmaiden Protocol performs across a variety of Apple hardware and macOS versions. Testing was conducted on both Apple Silicon (M1, M2, M3) and Intel-based systems, spanning macOS Monterey, Ventura, and Sonoma. Results demonstrated that Apple Silicon devices consistently exhibited faster reaction times and lower CPU impact, averaging under 1% during idle runtime.

Notably, the MacBook Air with an M3 chip on Sonoma 14.1 achieved the lowest latency and highest efficiency, while Intel-based machines showed slightly delayed execution of shutdown and sleep commands. Cross-platform compatibility was verified, though performance and detection precision varied slightly with older macOS builds.

An experimental bridge test with iOS devices revealed partial detection capabilities when tethered, highlighting potential for future cross-device support. Overall, the protocol maintained functional integrity in diverse operating environments with minimal resource overhead.

Environmental Condition Table

Platform	macOS Version	Silicon	Result Summary	Performance Impact
MacBook Pro 2023	Ventura 13.5	M2	All scenarios detected except BLE	<1% CPU
Mac Mini 2020	Ventura 13.2	Intel	Slight delay in shutdown (<2.4s)	2% CPU peak
MacBook Air 2024	Sonoma 14.1	M3	Full compatibility, lowest latency	<0.8% CPU
iMac 2019	Monterey 12.6	Intel	All USB events caught; slow sleep command	~3% CPU
iPhone 15 Pro (test via macOS bridge)	iOS 17.5	A17	Partial detection (USB over debug)	



The graph above is showing the reaction time and CPU impact of the Rigmaiden Protocol across different Apple devices and platforms based on your Environmental Condition Table.

Niche environments for "pressured" Rigmaiden to preform

Trigger Type	Response Command	Description
USB Vendor ID mismatch	shutdown -h now	Full system power-off
Unauthorized network interface	pmset sleepnow	Instant sleep to memory
BLE interface change	None (blind spot)	To be patched
Keyboard input rate anomaly	osascript + Lock	Simulates panic button
Unknown serial device	pmset displaysleepnow	Obfuscate screen instantly

Here's a representative equation that models Rigmaiden's response latency **(R_t)** as a function of system poll interval (P), hardware detection delay (D), and command execution time .

10. Conclusion

Rigmaiden introduces a novel and replicable model for USB/network surveillance detection and response. By empowering the user with a kill-switch governed by minimal but effective logic, it adds a last line of defense where traditional antivirus or MDM agents cannot reach. This approach bridges a practical gap in field security without relying on proprietary hardware.

11. Citations

Mendy, Michael (2025). Rigmaiden: A Protocol and Anti-Forensic Defense Tool Against Hardware Surveillance on Apple Platforms. https://github.com/Montana/rigmaiden

Please contact the author Michael Mendy at michael@rigmaiden.sh