MATTHEW R. WALSH
19197 GOLDEN VALLEY RD #333
SANTA CLARITA, CA 91387
(661) 644-0012

Plaintiff In Pro Per,

UNITED STATES DISTRICT COURT

CENTRAL DISTRICT OF CALIFORNIA

MATTHEW R. WALSH Plaintiff In Pro Per, vs. ROKOKO ELECTRONICS (AND DOES 1 THROUGH 50, INCLUSIVE) Defendant	Case No.: 2:25-CV-05340-ODW-RAO Before: Hon. Otis D. Wright II Courtroom 5D **EXPERT REPORT and Testimony of Matthew R. Walsh**

1. Expert Identification

I am Matthew R. Walsh, I submit this report as the primary expert witness in this case. While I am the Plaintiff in this matter appearing pro se, I am competently providing this document as a wholly neutral expert witness report. This report is based solely on the evidence before me and the factual basis surrounding that evidence.

It should be stated that I am not the expert witness in this action *because I am the Plaintiff,* but rather, I am the Plaintiff in this action because I am the most

qualified person to litigate this matter due to it's highly technical nature. From past experience, hiring an attorney simply adds a layer of dilution between myself and the Court which I believe slows or prohibits the technical flow of information in this case; therefore I must act as both the litigator and the Plaintiff. This report is submitted wholly as a neutral forensic investigator; and purely of technical analyses and finding devoid of argument. The evidence has all been authenticated under penalty of perjury and provided to the Defendant prior to the authoring of this document.

I submit this declaration in support of the Plaintiff, Matthew R. Walsh's action against Defendant, Rokoko Electronics.

2. Qualifications

I have over 26 years of experience in computer programming, software engineering and architecture design and development; and in computer security. I am a digital investigator and forensic analyst presently retained by law firms including but not limited to Harder Stonerock, LLP; and have been an expert in high profile cybersecurity/computer hacking/CFAA adjacent-Court cases dating to 2008.

As part of my daily profession, I am a professional software engineer systems architect where I solely engineer and program highly complex systems (700,000+ lines of code) using more than twenty (20) programming and scripting languages for both the private and government sector – including LA County and the Department of Homeland Security. I am wholly self-employed by my own software company which produces international industrial software and automation systems, established in 2007.

I have authored published security tools beginning at age 14 that have been recognized by Microsoft, McAfee, Trend Micro, Symantec, Sophos, and others, and my work has been featured in multiple magazines. I became a full time professional programmer at 15 and enrolled in evening college for Computer Science at 16 years old which I achieved a 4.0gpa. My professional career since then has experienced no gaps and I have worked on hundreds of projects.

Further, I am highly experienced in video game/film production. I am a published video game developer with titles(s) licensed by Nintendo, Sony and Valve. I own and operate Winteryear Studios, a game, film, 3D, VR, animation studio. I am qualified in all aspects of production including expert knowledge of motion capture, facial recognition and deep fake technology.

Lastly, I have experience in commercial-scale microcontroller-based (ARM, ESP, Arduino, Basic Stamp, etc.) hardware and software development, including

the authoring of custom protocols for communication and execution stacks and firmware development.

3. Assignment and Scope

The causation chain of why I became involved is quite simple.

Hardware Failed → Company Refused Repair → Engineer Investigates

As Plaintiff's Rokoko hardware no longer worked and Rokoko had refused parts, repair and replacement for nearly seven months; I chose in April of 2025 to use my technical capabilities to self-diagnose the equipment hoping to restore it's functionality and continue development of Plaintiff's video game The Next World. Instead, during my analysis, I discovered numerous issues which are the sole basis of this lawsuit and highly detailed within this document.

4. Materials Reviewed

This report is based on **(a)** Rokoko Studio, it's underlying communication layers, encrypted communications, programmatic requests/responses/replies to their servers and cloud application layers and system logs. **(b)** Rokoko Smartsuit, it's firmware, sensors, system logs, hardware components, communications with

my own computer and Rokoko's own internal developer tools **(c)** Rokoko's format of animation data files **(d)** Real-life events which occurred and in relation to these technologies **(e)** Rokoko's own statements.

All materials are present in produced discovery in which the Defendant has possession of nearly 3,750 pages of evidence. Throughout this document, it will be referenced by the bates stamp which corresponds to that evidence (e.g., WALSH000123).

5. Technical Background

The Rokoko motion capture system is based on a number of interdependent components, both physical and software.

1. **("The Textile") -** a clothing garment which houses most of the hardware. It is specific clothing designed with channels to hide the sensors and the Trunk wiring for each limb.
2. **("The Hub")** – a standalone computer system mounted on the users lower back which interprets the sensor data and transmits it to the Rokoko Studio software using TCP/IP over WiFi.
    a. The Hub is powered by what appears to be a low power Redpine RS9113 ARM Cortex semiconductor. The Hub is complete with

WiFi, USB and Device I/O for connection to the sensors as the Redpine chip comes standard with those stacks integrated (Wi-Fi, BT, BLE, ZigBee and Thread).

b. The sensors branch from the hub using six wiring trunks (aka "wire harnesses"), one for each limb and one for the head/neck.

c. Once a trunk reaches where a sensor is located, the trunk stops, connects to the sensor, then on the opposite end another **("Wire")** trunk continues downward until it reaches another sensor and continues the pattern.

d. The Hub has a core software structure containing the firmware, bootloader, unique_id (in this format: xx-xx-xx-xx), usb_api_version, wifi_api_version, bt_api_version (Bluetooth), serial_number (12 byte length), latititude, longitude, earth_mag_field_x (magnetic), earth_mag_field_y, and gravity.

3. **("Sensors") –** the sensors track inertial and kinetic movement changes and report them back to The Hub at fixed polling intervals (~100 fps).

i. The Code references the following sensor data vectors: First_buffer1, First_buffer2 First_buffer3, First_buffer4, signature, version (format: 00.00.00-00), [**boot_status],**

boot_version, is_calibrated, epoch, calibrationID, id_unique, reserved, SerialNumber (byte format: xx-xx-xx-xx)

b. Rokoko states the sensors have four LED states:

i. No color - Booted up properly.

ii. Red – Sensor is not working properly.

iii. Blue – Communication with The Hub cannot be established (faulty cable)

iv. Blinking Green – Sensors in bootloader mode; the most important status to recognize for the matter at hand.

4. **("SmartGloves") –** Textile gloves which are placed onto the users hands and connected by USB to a battery source. They contain their own miniature onboard computer which acts the same as The Hub and have smaller sensors for finger joints. The gloves do **not** physically connect to The suit's Hub, nor do they use 'The Wires'.

5. **("The Firmware")** – The operating software for The Hub, The Smartgloves and the Sensors. It is a single .BIN file with a RKKSFO (**R**o**K**o**K**o **S**an **F**ranci**s**c**O)** binary signature which is a large, multi-container bucket with an enveloped header, trailer and payload, which carries about six separate firmware files. The firmware is downloaded programmatically by Rokoko

Studio from the URL (https://cdn.rokoko.com/firmware/) and then installed onto the SmartSuit and SmartGloves hardware.

6. According to analyses of Rokoko's own source code, a single firmware package holds multiple upgrade targets: (a) UpgradeHubFirmware (b) UpgradeSensorFirmware (c) UpgradeModulesFirmware (d) UpgradeBootUpgrader (e) UpgradeBootloader (f) UpgradeSensorFromBoot

    a. The firmware extracts a number of individual firmware payloads. The overall structure references a key, SerialNumber and Status; while the payloads are: [SmartGlovesFirmware, HubFirmware, SensorFirmware, RedPineFirmware, BootLoaderUpgrade and BootLoader] indicating full ecosystem hardware modification upon firmware update.

    b. The firmware update system contains default hardcoded firmware files; assumably the lowest allowed baseline versions such as smartsuit_pro_sensor_firmware_v2.0.1-18-release.sfs, smartsuit_pro_hub_firmware_v2.6.3-81-release.sfh, smartsuit_pro_hub_firmware_v2.5.4-950-release.sfh, smartsuit_pro_sensor_firmware_v2.0.0-1277-release.sfs

7. **("Rokoko Studio") or Rokoko Studio ("Legacy") –** The software which runs on a users computer and connects to Rokoko Hardware (up to 10

SmartSuit's at a time) and servers. It is responsible for (a) managing the scene; actors and equipment (b) receiving and recording the animation data from each users' suit (c) translating the raw vector and inertial data into spatial 3d animation data (d) providing visualization of that data and allowing it to be refined by the user (e) enabling data import and export functionality and (f) handling device health, diagnostics and firmware updates.

8. **("Animation Data")** or ("Animation") or ("Intellectual Property") – The movements from the Sensors are read by The Hub and transferred to Rokoko Studio Software which then translates them into spatial 3D movement at a fixed framerate interval; capturing the actors movements almost perfectly and recording them.

    a. The Animation Data for Legacy is stored in a .srec file with an accompanying .srec.meta file. The .srec file has a binary signature of "RKK!SR".

    b. Most importantly, **these file(s) clearly contain CMI** including the 12-byte serial numbers of each actors Hub, the actors name, height, biometric measurements and the serial numbers of each Sensor.

9. **Rokoko ("Teams")** – A paid, subscription based, opt-in service which collects user animation data and stores it in an online cloud platform on Rokoko's U.S. based servers.

10. **("System Files")** – Rokoko software uses a number of file formats, Recordings (.srec), Project Container (.ssproj), Profile (.ssprof), Project File (.ss), SmartSuit Firmware (.sfh), Hub Firmware (.sfh), Bootloader Firmware (.sbh), Sensor Firmware (.sfs), WiFi Configuration (.wifi)

11. **("Rokoko Servers")** – the remote internet-reachable servers that Rokoko utilizes in their operations. All U.S. based. This is where Rokoko Studio communicates with Rokoko's internal software, api layers, databases and stores account information, customer information, animation data, users intellectual property and more.

    a. ping.rokoko.com [18.65.3.76] amazon San Francisco
        i. This appears to be a user tracking / phone home system.
    b. cdn-studio.rokoko.com [3.169.252.38] amazon San Fran.
        i. This appears to be Rokoko Studio's main operational server which hosts various downloads (such as Rokoko Studio, firmware, etc.)
    c. fw-api.rokoko.com [13.226.225.121] amazon San Francisco

i. This appears to be the firmware API server which handles firmware requests, matching and downloads.

d. rmp-gql-public.rokoko.com [3.167.192.77] amazon San Francisco

i. This appears to be Rokoko's GraphQL server for bidirectional data transfer.

e. id.rokoko.com [13.52.115.166] amazon San Francisco

i. This appears to be Rokoko's user authentication server.

f. rokoko-id-new.netlify.app [54.215.62.21] amazon San Francisco

i. This appears to be a third party API building tool.

g. rmp-team-gql.rokoko.com [3.167.212.100] amazon San Fran.

i. This appears to be Rokoko 'Teams' GraphQL server

h. cdn-scene.rokoko.com [18.164.174.97] amazon L.A

i. This appears to be Rokoko's remote scene image rendering platform (turns 3d into 2d images)

12. **("Rokoko Database")** – The database system and query language which runs on Rokoko's servers and acts as a data storage and intermediary query language between Rokoko Studio and Rokoko's Servers and Database. The query language used is Facebook/Meta's GraphQL for data, as well as WebSocket PUT requests for file transfer. The GraphQL language and

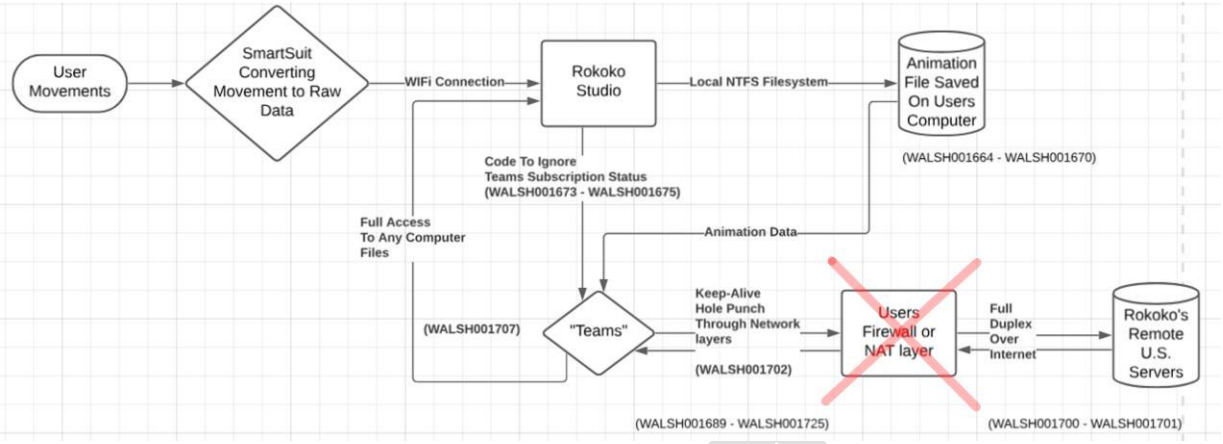
WebSocket PUT requests match and authenticate the described and visual evidence used throughout this action.

6. Motion Capture Production Pipeline

The animations at issue were produced through a very standard workflow pipeline involving motion capture recording, data processing, export to animation files, and integration into the game engine.

Screenplay → Live Acting → Actor Movement → Sensors → Hub → Network → Computer → Rokoko Studio → Animation File → *(Rokoko Harvesting Data Secretly)* → Human Animator for Manual Cleanup → Retargeting for Game Character → Video Game → Sequencer or Animation Blueprint → Complete

Illustration:



7. Firmware Behavior and System Failure

Analysis of the firmware updates and system behavior demonstrates that Rokoko developed and then released a very specific firmware update, which the developer notes indicate **they were knowledgeable of the fact that it would permanently destroy gen1 hardware right around Rokoko's End of Life for that unit (September, 2024).**

First, Rokoko authored and thereby created The Firmware which had a 319 kilobyte filesize. **Second,** Rokoko published The Firmware to their servers on January, 9, 2024 as version 7.2.3.0-94. **Third,** At a pricely unknown time in 2024, Rokoko set the release flag to "1" which in their system which is a standard boolean signal to tell one or more users' Rokoko Studio software that it is time to download and install The Firmware.

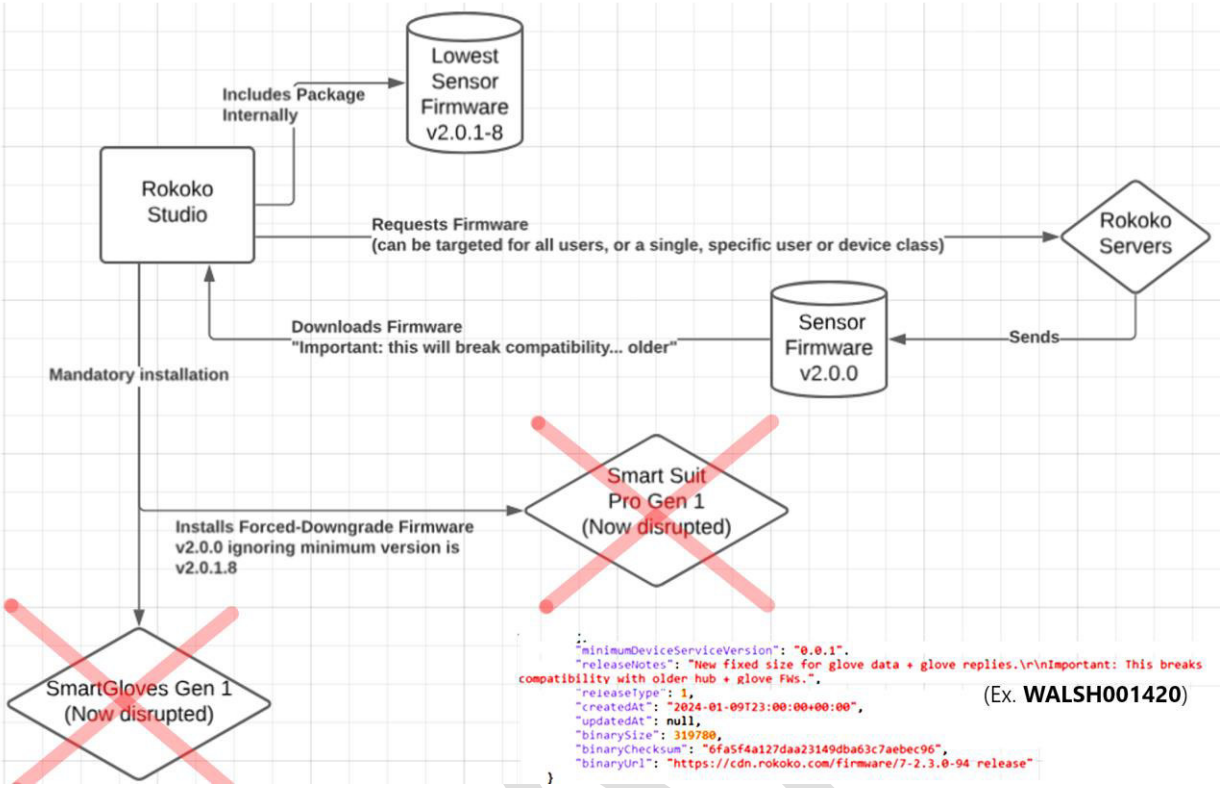
Most importantly, the developer release note as hosted by Rokoko's servers stated: *"Important: This breaks compatibility with older hub + glove*

[Firmwares]". As of April 2025, when much of this investigation was conducted, this appeared to be the most recent / last released firmware by Rokoko for the Gen 1 SmartSuit Pro, such as Plaintiffs'; indicating it was, in fact, the operative firmware which affected the Plaintiff's equipment.

Of significant import, Rokoko Studio's source code shows the default, lowest sensor firmware allowed, provided and auto-installed is "smartsuit_pro_sensor_firmware_**v2.0.1-18**-release.sfs"; however, in Plaintiff's logs and diagnostic reports post-upgrade – the sensor firmware version is much lower than the minimum viable baseline -- **v2.0.0.0.**

It should be noted further that the developer notes openly indicate that it only *"breaks compatibility"* with "[*older] hub + glove"*. It is well within held technical knowledge – and further explained here in in plain English -- that the firmware targets both the old hardware and the new at the same time without distinction. The Firmware clearly states it would disrupt the operation of only Plaintiff's Gen 1 hardware, while their newer Gen 2 hardware would continue operationally.

Illustration:



8. Supporting Technical Evidence

This conclusion is supported by system logs, developer records, network captures, software behavior observations, Defendants' source code, and related diagnostic materials along with the Defendants' public statements. The tools used are industry standard and household names to those who engage in this field of work. The tools, methodologies and procedures meet the standards of Daubert, as I myself do as well.

1. **Firmware and Hardware Failures:**

   (Bates No. WALSH001420 - WALSH001468)

2. **The First Disruptive Firmware Release and Rollback -** In 2023, Rokoko released a firmware update which crippled both SmartSuit units and the SmartGloves. The prior day all equipment worked. After The Firmware update, neither did. As of 2026, Rokoko's website presently states that The Firmware had in fact, caused these very issues.
3. Rokoko at this time, as is demonstrated by the evidence, admitted directly to the Plaintiff that the firmware had failed (3/20/2023).
4. The suits remained wholly disabled pending Rokoko customer service intervention which appears to have taken 30 to 60 days for each engagement.
5. Rokoko requested log files from the SmartSuit to determine the cause of error and required remote support sessions in which Rokoko interacted with Plaintiff's computer system directly.
6. Rokoko support placed their internal customer support tools on Plaintiff's computer. Once they were finished, they right clicked the tools and sent them to the Recycle Bin, but did not permanently delete them.
7. Rokoko support determined the suit had physical damage with 'wires', and sold replacements. The 'wires', once again, are wire harness/trunk extensions which go from sensor to sensor, daisy chaining them.

8. **Manufactured Conflict of Fact -** It should be noted, that the LED states on the sensors, contrary to Rokoko's own support documents were blinking green; as reported to Rokoko multiple times. This indicated the sensors were stuck in bootloader mode **not having communication issues with the hub or wiring** (which would instead would cause the sensor LED's to illuminate Blue).
9. Upon changing the wires, the suits remained non-functional, the sensors still in blinking green LED bootloader mode. Rokoko's suggestion was to send the suits in for repair. As Plaintiff was deep in production at the time, other animation solutions were utilized.
10. **Self-Repair of the Suits -** Upon realizing Rokoko's official developer tools were still on Plaintiff's computer, they were removed from the Recycle Bin and used to re-flash the SmartSuit's to an earlier firmware file. One suit began working again, the other remained non-functional as it would not accept a new firmware file. **This confirmed the firmware had been the issue, not the wiring.**
11. There were hundreds of required scenes which were single-person only; having only one suit at this time was of no consequence.

12. **Final Suit-Destroying Firmware -** Months later, Rokoko issued a mandatory firmware update. This firmware immediately caused the Rokoko Smartsuit Hub and Sensors to not operate.

    a. Within Rokoko code (FirmwareUpdateModal.cs), a Boolean is present DISABLE_FIRMWARE_UPDATES which by default is set to true; indicating Rokoko retains the power to remotely trigger these updates and make them mandatory, despite their public website stating all updates are optional.

13. The lights on the sensors showed the sensors were receiving power, transmitting and receiving data, and were not in a definitive fail state.

14. Log files show the hub began reporting technical failures and further that all sensors could not be detected, yet, were reporting errors.

15. Analyses of the sensors in both Rokoko Studio and on the sensor LED's showed they were in mixed boot states. Despite this, Rokoko stated their interpretation of the logs stated the Wires indicated failure, however, the logs are plain text and obvious in language and they clearly indicate sensor and hub failures. Further, the SmartGloves as well no longer worked, which was telling – as the gloves do **not** connect to The Hub, nor do they use 'Wires' for connectivity.

16. According to the evidence, Rokoko denied parts, repair or replacement for a period of seven months (September 2024 to April 2025). Finally in about March, 2025; Rokoko once again requested log files stating the log files they had were 'too old'. Plaintiff complied, and once again, Rokoko stated the log file indicates wiring issues. The log files, in fact, did not show wiring issues – they showed sensor and hub failures. Precisely matching Rokoko's developer note warning.

17. **Forensic Analyses of Rokoko Systems:**

18. In April, 2025; I began forensically investigating the Rokoko Studio software. To do so, I needed to intercept their communications and data.

    a. **Protocol Discovery**

    b. I first used the industry standard tool WireShark, a packet sniffer, to detect and monitor the traffic coming and going from Plaintiff's computer. By default, WireShark returns all communications that occur from all programs. I continued filtering and isolating the traffic until it was narrowed to only communications between Plaintiff's computer and Rokoko's servers.

    c. **Rerouting Communication Traffic, Bypassing Encryption**

    d. Now knowing the IP addresses, ports, protocols, encryption methods and endpoints, I then used a tool called Proxifier which allows for

select programs to have their communications temporarily redirected to a different endpoint.

e. The communications between Plaintiff's computer and Rokoko's servers was clearly occurring over TLS (Transport Security Layer) encrypted by SSL (Secure Socket Layer) communicating via HTTPS (Hyper Text Transport Protocol; port 443)). To overcome this encrypted communication stack, I installed a new Root Certificate Authority (CA) ("Fake SSL certificate") into Plaintiff's system's "Trusted Root Certification Authorities" store. This creates a secure communication layer that *appears* secure, however, as there are no unknowns, all data that flows can be easily intercepted and monitored.

f. Instead of allowing the communications of Rokoko Studio to go directly to Rokoko's servers in a 1:1 exchange, I used Proxifier to instead route the traffic from Rokoko Studio to MITMProxy to Rokoko's Servers.

g. **Passive Man-In-The-Middle Attack**

h. I then routed the Proxifier data to MITMproxy, a widely used security monitoring tool.

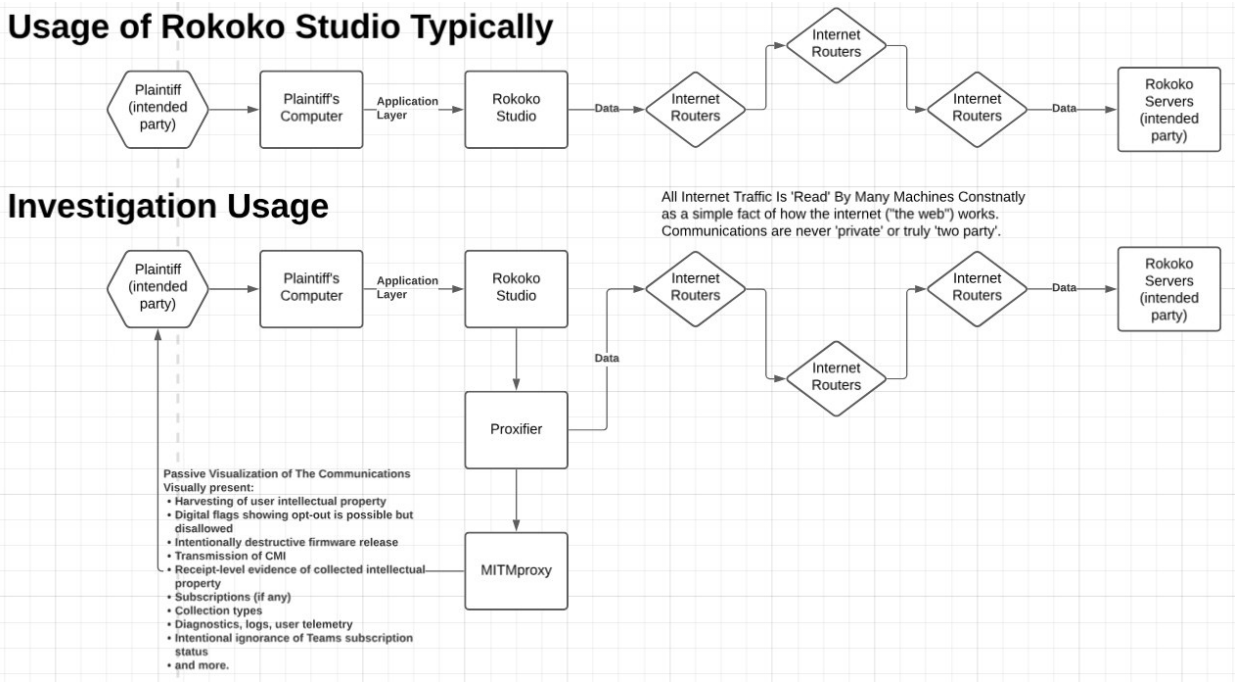
i. Using this stack of tools, I then performed a MITM (Man-In-The-Middle-Attack) which is essentially where there is normally two

endpoints: Rokoko Studio (local on Plaintiff's computer) to Rokoko Servers (remote/internet); but instead, now, there are three endpoints: Rokoko Studio (my computer) to MITMproxy (my computer) then to Rokoko's Servers (remote/internet).

19. **Legality**

It is important to state that while these methods may appear unscrupulous, the CFAA (Computer Fraud and Abuse Act) as well as international data and privacy and computer hacking laws only apply to situations where the interceptor is not an authorized party to the communications. Here, there is no lack of privilege to the communication as Plaintiff is an intended recipient of the communications as an authorized user of the Rokoko Studio software. All communications are two-party between the Plaintiff and Rokoko, of course. As a matter of fact -- due to how the internet works -- all communications are inherently intercepted potentially dozens or hundreds of times and passively monitored before reaching their destination. Further, all interception was only performed on software and communications contained on Plaintiff's own computer system. As one of two parties to the communication, Plaintiff has a lawful right to be a party that communication. (Van Buren v. United States, 593 US 374 (2021))

Illustration:



20. **Analysis and Production of Evidence**

(Bates No. WALSH002407 - WALSH002415)

(Bates No. WALSH001664 - WALSH001725)

21. The first step to bringing a claim for DMCA violations was to assess whether the .srec Animation files contained CMI or not. They were opened in HxD, an industry standard Hex Editor (binary file & memory viewer/forensic analyses tool) widely used for more than 20 years.

a. The animation files did in fact contain CMI (WALSH001664 - WALSH001670) including but not limited to biometrics, author name, actor names, unique ID's, device serial numbers, etc.

22. Analyses of the underlying communications provided most of the underlying evidentiary matter in this action:

a. **Rokoko Intentionally Released Destructive Firmware** – The intercepted communications from Rokoko's server provided a list of firmware packages which have been released in JSON format. The most recent firmware contained a developer note which said *"Important: will break compatibility with older glove + hub [firmwares]"*. This firmware was released to the public despite this knowledge.

i. It is worth noting that the wording here is specific, it does not break compatibility with *newer* gen 2 devices, only *older* gen 1 devices. Meaning, gen 2 will be the only operable devices and gen 2 will be 'destroyed' upon release.

b. Rokoko consistently received live diagnostics and knew of the firmware's damage – Rokoko did not need logs, nor customer support sessions or access to Plaintiff's computer. They were always

made aware, instantly what issues the SmartSuit hub and sensors were experiencing as detailed below.

c. **Rokoko knew the sensors were malfunctioning and the firmware was installed** – The live diagnostics clearly indicated the firmware version of the sensors and The Hub. The diagnostics also clearly stated the sensors had failed, were stuck in mixed boot states and could not start. Despite this, they continually suggested 'wires' for two years. This JSON diagnostic envelope is sent live to Rokoko consistently:

.. {"device": "smartsuit_pro", "device_serial_number":"R8PZA06KVJYU" …. **"successfully_initialized":false, "device_sensor_firmware":"2.0.0-r", "has_error":true, "error_type":"sensors_in_mixed_boot_states"}, …**

i. It should be noted, the sensor firmware version is NOT the same as the firmware version which Studio downloads. As stated prior, there are multiple firmware packages inside the

single .bin file which has it's own version release number.

d. **Rokoko harvested my intellectual property** – Plaintiff's intellectual property was immediately transferred to Rokoko servers over WebSocket PUT requests.

e. Creating new animations or even copying existing ones to new names such as 'WILLROKOKOTAKETHIS' and 'PLEASEDONTSYNC' immediately caused those animations to be taken by Rokoko, who then programmatically provided a confirmation of collection, storage and receipt.

23. **Analyses of Rokoko's Source Code**

24. I further investigated the Rokoko Studio software by obtaining the Source Code to the software and it's underlying systems and dependencies.

a. Rokoko Studio utilizes Unity (a video game engine).

b. As Unity's programming language of choice is C#, that means that all software written in C# is compiled to MSIL (**M**icrosoft **S**ymbolic **I**ntermediate **L**anguage)

c. I used industry standard applications .NET Reflector and JetBrains DotPeek to load the various Rokoko Studio assemblies, which revealed mostly unobfuscated source code.

i. **LEGALITY:** It should be noted that accessing MSIL code is not actually decompilation nor is it reverse-engineering. The code is stored in essentially comparable plain-text format (as symbolic bytecode) and is interpreted by the JIT (**J**ust **I**n **T**ime) Compiler (the executable package which in the code is packaged with) to create a running application. Rokoko did nothing to protect or hide the visibility of their source code.

1. They did not compile to Native Code (bypassing the JIT compiler and MSIL)
2. They did not obfuscate their code (a simple, even free and standard programmatic scrambling process)
3. Rokoko simply provided their code in a format which could both be executed and used *and read in it's original source code format*.

d. The source code shows a number of issues:

i. **Forced Consent of Terms -** Rokoko forces that the user has accepted the Terms & Conditions (EULA) even in instances where they have not.

ii. **Username/Password Bypass Backdoor -** Rokoko Studio contains a backdoor written by Menalos (an ex-employee from

three years ago in Greece) who wrote an authentication bypass to allow him (or anyone with knowledge) to log into Rokoko Studio without the users username or password and then have unbridled and untethered access to Rokoko Studio which contains intellectual property, animation data, user account information and so forth.

iii. **Harvesting Intellectual Property By Ignoring User Preference -** The 'Teams' system ignores whether a 'Teams' subscription is in place and instead forces collection of intellectual property, regardless of subscription, payment or user consent. In Plaintiff's case: Plainitff never signed up for Teams, never added a collaborator, a credit card, was never billed, did not approve and yet the Rokoko Studio software was programmed to ignore that and still collect the intellectual property. The intercepted communications confirmed all of this.

iv. The intellectual property, when harvested is taken over WebSockets and transferred to Rokoko's servers.

v. **Harvested Intellectual Property Targets a 'Public' Directory** – The code shows that the intellectual property is

targeted to a folder called /Public with no further designation of project, user, etc.

vi. **Exceeding Access To Users' Computer Systems And Bypassing Network Security Layers -** Strangely, the 'Teams' code also creates a webserver on the users computer.

vii. The webserver binds to * (all adapters/IP addresses/networks) and creates a listener.

viii. It further uses MQTT (an "uninterruptable" lightweight messaging protocol developed by IBM) along with a hole-punch mechanism (creates a hole through a multi-NAT (Network Address Translation) network so that Rokoko can connect inward and bypass security restrictions like firewalls).

ix. The WebServer operates in a Default Realm of 'SECRET AREA'.

x. This system also included a keep-alive, which is a timed outbound connection "pulse" at a fixed interval whose goal is to never allow the hole-punch (or pinhole) to close.

1. It is worth noting, for a system which uses WebSockets to PUSH data to Rokoko, it is concerning that Rokoko

creates a server and a security bypassing pathway **inward, from Rokoko into the users computer.**

xi. **Rokoko Claims GDPR Compliance, Has No European Servers –** Rokoko Studio code confirms all of Rokoko's operational servers are in America (Downtown Los Angeles), AWS in Virginia and Washington). Yet, Rokoko claims on their website they comply with GDPR privacy and data regulations; and even holds many government contracts with the Danish government in which they state they comply with GDPR.

xii. **Overreaching File I/O Exposed To Backdoor Access –** The webserver that Rokoko Studio initiates gives anyone who can connect to it full access to read any file on the computer with no restrictions or safeguards:

```
public byte[] GetFile(string path) {
        return File.ReadAllBytes(path);
}
```

Additional, there are registered events for expected inbound activity: OnGet, OnPost, OnPut, OnDelete, OnPatch,

OnConnect

The server spins up a thread (essentially a program within a program which does not interfere with the main system) and is actively listening for inbound connections:

this._listener.Start();

this._receiveThread = new Thread(receiveRequest);

xiii. Rokoko Studio already transmits Animation data to Rokoko servers using WebSocket-based file PUT's; which is full duplex communication. Therefore, as the file sharing mechanisms already existed in place, I could find no legitimate use for this functionality, which bypasses all network layers and firewalls to provide access to users computer systems available to the outside world.

9. **Beyond Technical Evidence**

1. Rokoko's website and materials make clear, plaintext express admissions to the allegations set forth in this report.

a. 2022 pre-investor pitch deck outlines plan to harvest, collect, resell, sublicense and use users' intellectual property (Bates No. WALSH001727- WALSH001736)

b. The Website openly states that firmware updates have caused the above described issues and as of 2026 they are still requiring users to "contact Support for help to restore the sensors" and that "we're working on getting this properly restorable from within Studio" (Bates No. WALSH001446)

c. Claims of GDPR despite U.S.-only servers for collection and storage *(Bates No. WALSH001676, WALSH001700 - WALSH001701)*

d. Dataset made from users I.P. *(Bates No. WALSH001672)*

e. Policy and code force user reconnection and 'synching' (Bates No. WALSH001673 - WALSH001675); 'Enterprise' payment model required to prevent harvesting of Animation data (WALSH001737)

f. Animation I.P. will be collected, CMI stripped, resold, sublicensed *(Bates No. WALSH001680 , WALSH001682, WALSH001692-WALSH001693)*

g. Live website counter of collected user animations (Bates No. WALSH001690) updated daily.

h. Sublicensing of users I.P will occur (WALSH001691)

i. Users Animation data will be used to train Artificial Intelligence Models, of which Rokoko also sells a subscription to (Bates No. WALSH000003 - WALSH000008)

j. Video evidence from Rokoko's software shows harvesting of intellectual property in real-time (WALSH001679, WALSH001689, WALSH001699)

k. Terms & Conditions from 2020 – March 29, 2025 contain no license whatsoever for Rokoko to resell or sublicense users intellectual property. March 30, 2025 imposed sweeping retroactive terms back-claiming license (Bates No. WALSH002454 - WALSH002503) these agreements are packaged part and parcel with the software installation media (Bates No. WALSH002449 - WALSH002448).

10. **Reproducibility of Results**

1. The investigative procedures and analyses described in this report are reproducible by any qualified technical expert using the same commercially available tools and hardware/software environment. The tools referenced in this report -- including Wireshark, Proxifier, MITMproxy, HxD, JetBrains DotPeek, and .NET Reflector -- are standard industry tools widely used in cybersecurity, software engineering, and digital forensic analysis.

2. The following procedures may be repeated by an independent expert to reproduce the findings described herein:

3. Firmware and Hardware Failure Analysis:

    a. A Rokoko SmartSuit Pro (Gen1) system running the firmware version identified in this report may be tested by installing the referenced firmware update package and observing device initialization behavior through Rokoko Studio diagnostics and sensor LED states. System logs generated by Rokoko Studio will reflect the same error conditions described herein by the two devices analyzed, including sensors entering mixed boot states and initialization failures.

4. Firmware Rollback Verification:

    a. Using Rokoko's internal support tools referenced in the evidence, the SmartSuit hub and sensors may be flashed to an earlier firmware version. Where hardware has not been irreversibly affected, this rollback restores operational status of the suit, confirming that the failure condition was introduced by firmware rather than wiring or sensor hardware defects.

5. Network Communication Observation

    a. By running Rokoko Studio or Rokoko Legacy on a local system and monitoring network traffic using Wireshark, an expert may identify

the same network endpoints, protocols, and communication patterns between Rokoko Studio and Rokoko servers as described in this report.

6. Intercepted Communication Analysis

   a. By routing Rokoko Studio and Legacy network traffic through a local proxy using Proxifier and intercepting the traffic using MITMproxy with a locally installed trusted root certificate authority, an expert may observe the same diagnostic telemetry, firmware package listings, and animation upload communications described herein. It should be noted that Rokoko Studio and Legacy must be used extensively across all of it's features to achieve the same depth of evidentiary matter.

7. Animation File CMI Verification

   a. Opening Rokoko .srec animation files using any hex editor such as HxD will reveal embedded metadata fields containing Copyright Management Information (CMI), including device identifiers, actor metadata, and serial numbers consistent with the examples referenced in this report.

8. Software Behavior and Source Code Inspection

   a. Loading Rokoko Studio assemblies into .NET analysis tools such as JetBrains DotPeek or .NET Reflector allows inspection of the MSIL

contained within the assemblies, revealing the same programmatic behaviors and systems described in this report, including firmware update handling, data synchronization logic, and network communication structures.

9. Each of these procedures relies only on publicly available tools, the Rokoko software distributed to users, and the Rokoko hardware devices themselves. No proprietary Rokoko internal systems were accessed during this investigation. As such, a qualified expert with access to the same hardware and software environment will be able to reproduce the observations and results described herein unless Rokoko has taken post-litigation steps to alter, secure or prevent such access.

11. **Compensation**

I am the Plaintiff in this matter. As such, I have received no compensation for my testimony or expert opinion.

12. **Conclusion**

1. Based on my analysis, it is of my expert analysis and opinion that:
    a. Rokoko released a firmware update, which the developer notes clearly stated would ‘break’ gen 1 hardware. Once released, it rendered the

motion capture system unusable and destroyed my Rokoko SmartSuit and Gloves and all subcomponents therein.

b. In Rokoko Studio versions since at least 2022, Rokoko has harvested users intellectual property.

c. In all versions of the Rokoko software from at least January 1, 2020 to March 30, 2025 (including the March 29, 2025 version); all included license agreements shipped as part and parcel of the software never contained any license rights to users intellectual property.

d. As my hardware was destroyed in 2024, it is temporally impossible that I could consent to any such terms.

e. The conclusions as listed appear to be wholly intentional by design.

13. **Authentication of Evidence**

a. The evidentiary information relied upon in this report was generated, collected, and organized through an automated system designed authored and implemented by the undersigned. The system employs defined and repeatable processes for data acquisition, preservation, and output, including integrity checks at each stage. As the undersigned has full knowledge of the inner-workings of that system, these processes are sufficient to support a

finding that the resulting records are what they are claimed to be, consistent with Federal Rule of Evidence 901(b)(9).

b. The records comprising the evidentiary dataset exhibit distinctive characteristics, including consistent formatting, metadata, source identifiers, and internal relationships between files. These characteristics were analyzed for internal consistency across the dataset and are sufficient to support a finding of authenticity pursuant to Federal Rule of Evidence 901(b)(4).

c. The evidence relied upon in this report was further evaluated through comparative analysis, including re-testing and reproduction of results using independent methods. These comparisons yielded consistent outputs across formats, including demonstrative reproductions such as recorded testing procedures. This process supports authentication under Federal Rule of Evidence 901(b)(3).

d. The undersigned has personal knowledge of the methods used to collect, process, and analyze the evidence, including direct involvement in the acquisition and validation of the records. This knowledge supports authentication under Federal Rule of Evidence 901(b)(1).

e. All evidence referenced herein has been maintained in its original or forensically preserved form, without material alteration. To the extent any

format conversions were necessary for analysis or presentation, such processes preserved the substantive content of the records.

14. **Peer Reviewed Materials for Verification of Expert Assessments**

Please note: none of these materials were used to conduct the tests herein, nor were they relied upon for knowledge. They are being submitted post-authoring of this document simply to establish that the undersigned's own methods and personal knowledge are echoed in ISO certified, peer reviewed and published materials and scientific journals. All journals are ISO 9001:2008 certified) written by the International Research Journal of Engineering and Technology (IRJET).

1. **NAT Traversal and Hole Punching -** A thorough explanation of Hole punching across TCP, UDP and ICMP protocols has been provided to the Defendants in discovery.
2. **WebSocket Protocol for Real-Time Data Transfer –** A thorough explanation of real-time data transfers across WebSockets (how Defendant's allegedly transmitted intellectual property) has been provided to the Defendants in discovery.
3. **Cyber Attacks and Different Types (including MITM used here) -** A thorough explanation of various cyber attacks (which are only illegal / maleficent based on intent, and in fact, are just a regular part of daily software engineering and debugging processes).
4. **Research Paper on Machine Learning and its application-** A thorough explanation of machine learning and it's applications (one of the reasons for harvesting of intellectual property by the Defendants). This journal has been provided to the Defendants in discovery.
5. **An Artifact-Driven Framework for Detecting File Wiping and Timestamp Manipulation in NTFS Systems -** A thorough explanation of using the HxD tool (which Plaintiff used to analyze CMI within animation

files) and how it is used forensically. This journal has been provided to the Defendants in discovery.

6.

I declare under penalty of perjury under the laws of the United States of America that the foregoing is true and correct.

Executed March 8, 2026

Matthew R. Walsh
Plaintiff In Pro Per