Petlibro: Your Pet Feeder Is Feeding Data To Anyone Who Asks

December 27, 2025

BobDaHacker

They fixed most of it. But the authentication bypass that lets anyone login to any account? Still working after 2+ months. For the most popular smart pet feeder company. Cool.

What Happened

I was looking at the Petlibro app - they make smart pet feeders, water fountains, and other IoT pet products. Millions of pet owners use these things to feed their cats and dogs remotely.

What I found was... a lot.

The Vulnerabilities

1. Authentication Bypass - Login As Anyone

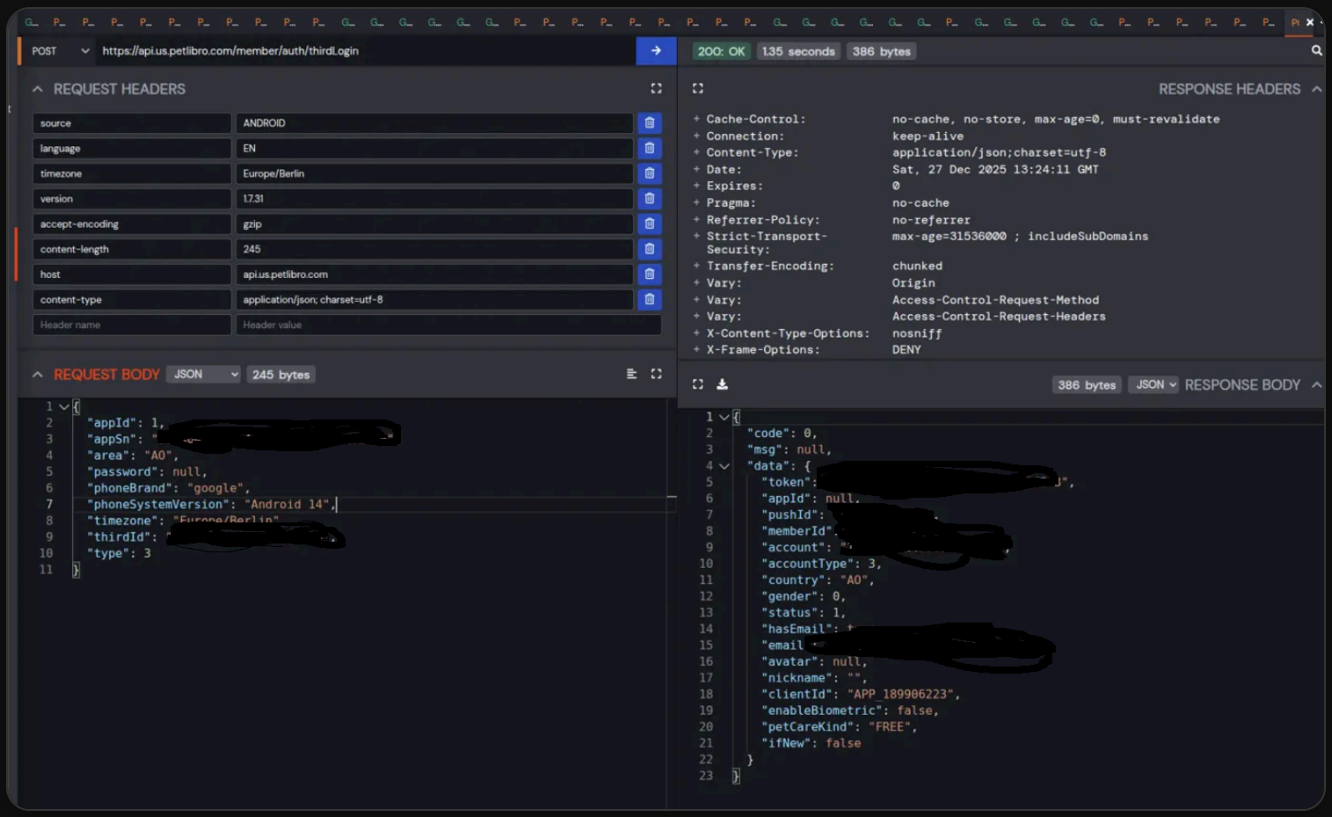
This is the big one. Their social login API at `/member/auth/thirdLogin` doesn't verify OAuth tokens. It just accepts an email and a Google ID directly from the client.

The request just needs:

```
⚡ terminal

{
  "phoneBrand": "google",
  "thirdId": "108305632482610644204",
  "type": 3
}
```

And you get back a full session token, member ID, email, and everything needed to access their account:



"But how do you get someone's Google ID?" Easy. Google's People API lets you look up anyone's Google ID from their email. It's public information - Google IDs are not secrets. That's exactly why you're supposed to verify the OAuth token server-side, not just trust a client-submitted ID.

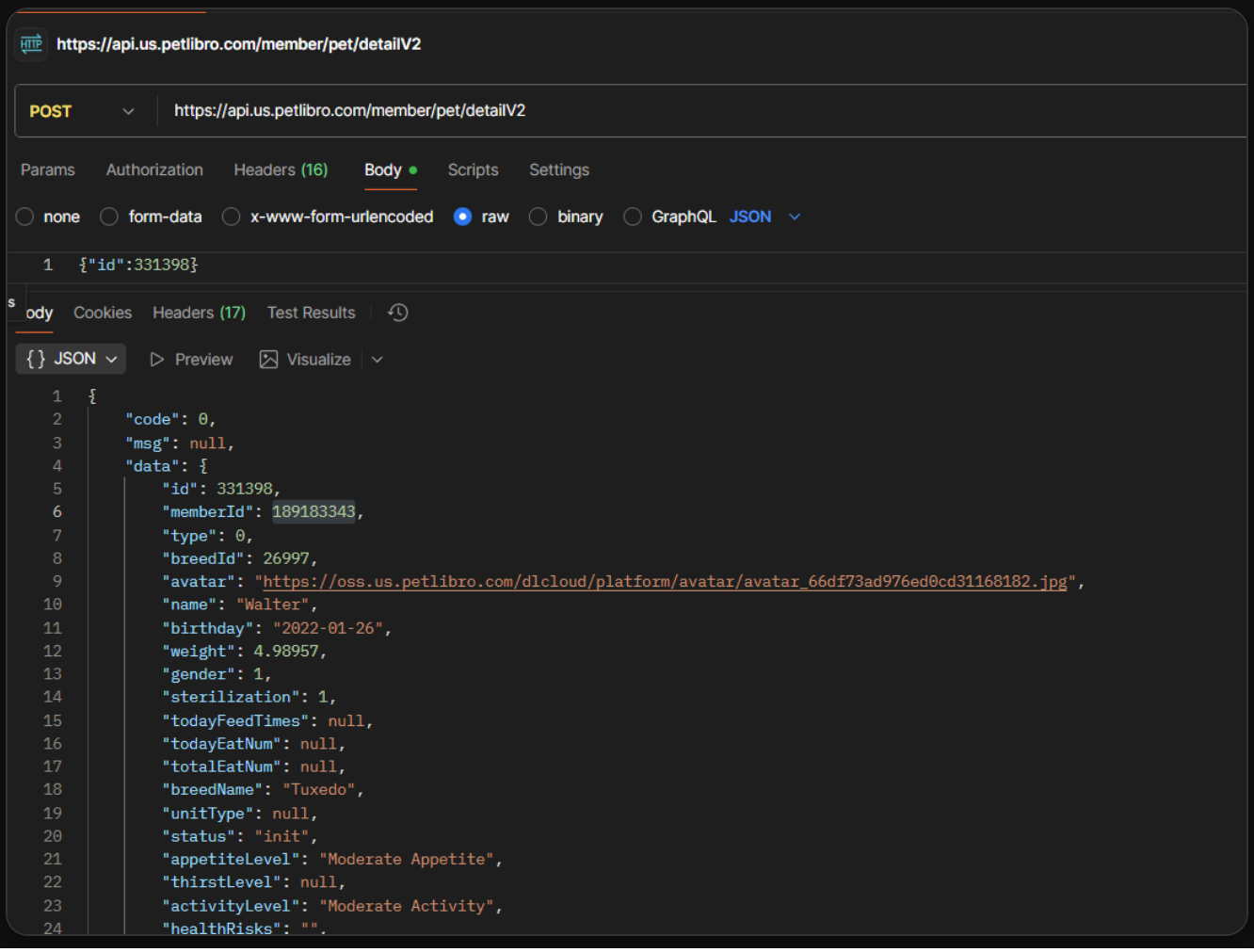
The entire authentication model is backwards. They're treating the Google ID like a password when it's essentially public information.

This is still working as of today. They made a new endpoint that properly verifies ID tokens, but left the old vulnerable endpoint active for "legacy compatibility" - they say they are waiting until their analytics show most users have upgraded before removing it. Two months later, anyone can still take over any account that uses Google login.

It's the Lovense situation all over again. "We have a fix but we're keeping the vulnerability active for legacy support." Your users' security should not be optional while you wait for upgrade metrics.

2. Access Anyone's Pet Data

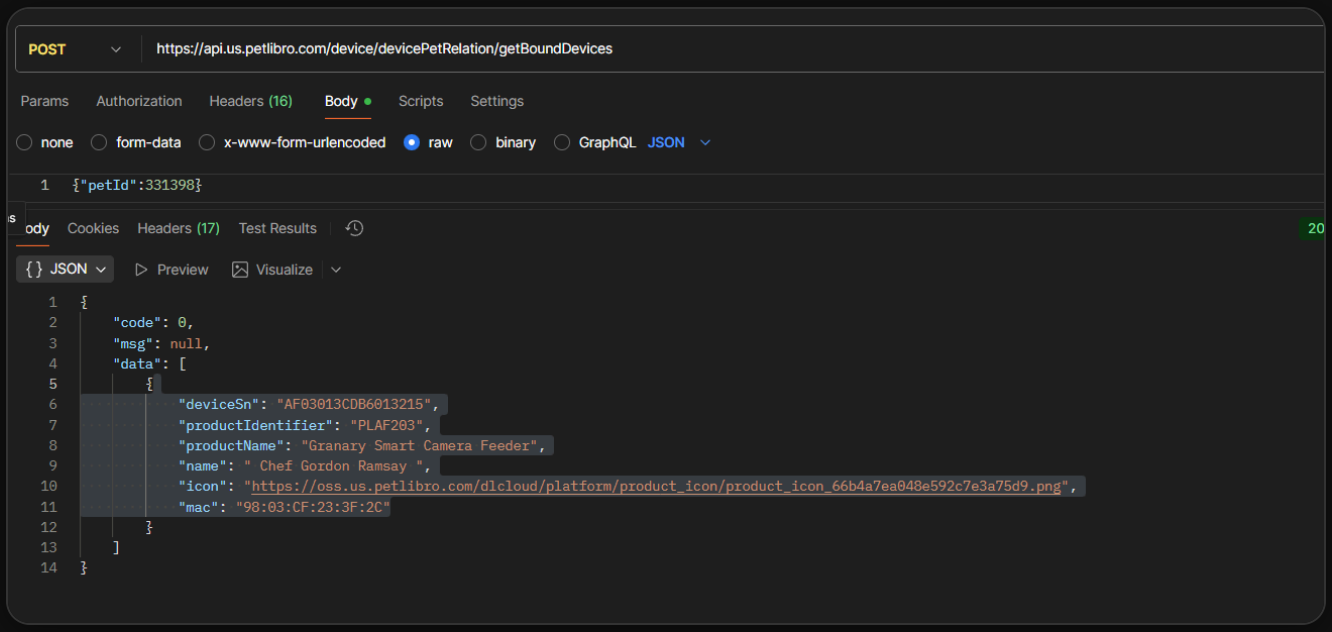
The pet detail endpoint `/member/pet/detailV2` doesn't check if you own the pet. Just send any pet ID:



Got a random person's pet named "Walter" - a Tuxedo cat born January 26, 2022, weighing 4.98kg, with "Moderate Appetite" and "Moderate Activity". Also got their avatar URL and member ID.

3. Get Device Serial Numbers and MAC Addresses

From the pet ID, you can get all bound devices via `/device/devicePetRelation/getBoundDevices`:



This returned:

- Device serial number: `AF03013CDB6013215`
- MAC address: `98:03:CF:23:3F:2C`
- Product name: "Granary Smart Camera Feeder"
- Device name: "Chef Gordon Ramsay" (lol)

With the serial number, you now have full control over that device.

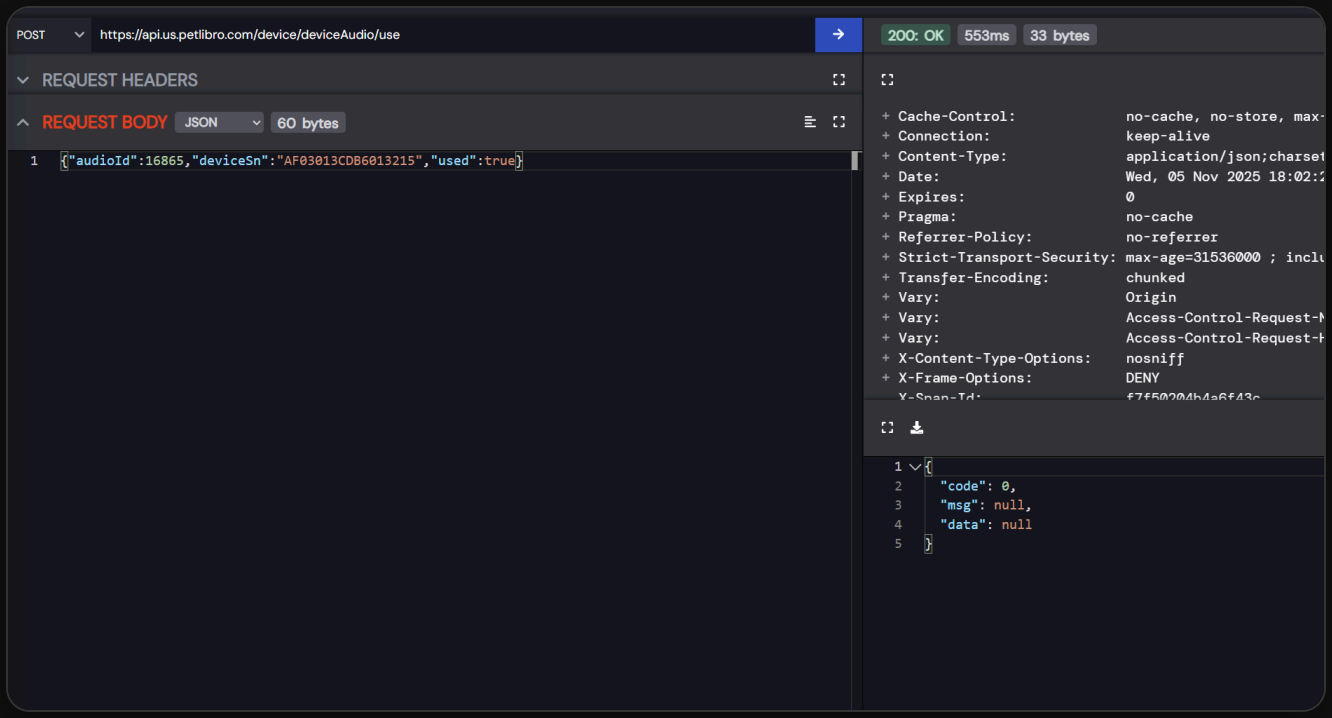
4. Hijack Any Device

ALL device APIs accept any serial number without checking ownership. You can:

- Change feeding schedules
- Trigger manual feeds
- Access camera feeds
- Modify device settings
- Basically anything the owner can do

5. Access Private Audio Recordings

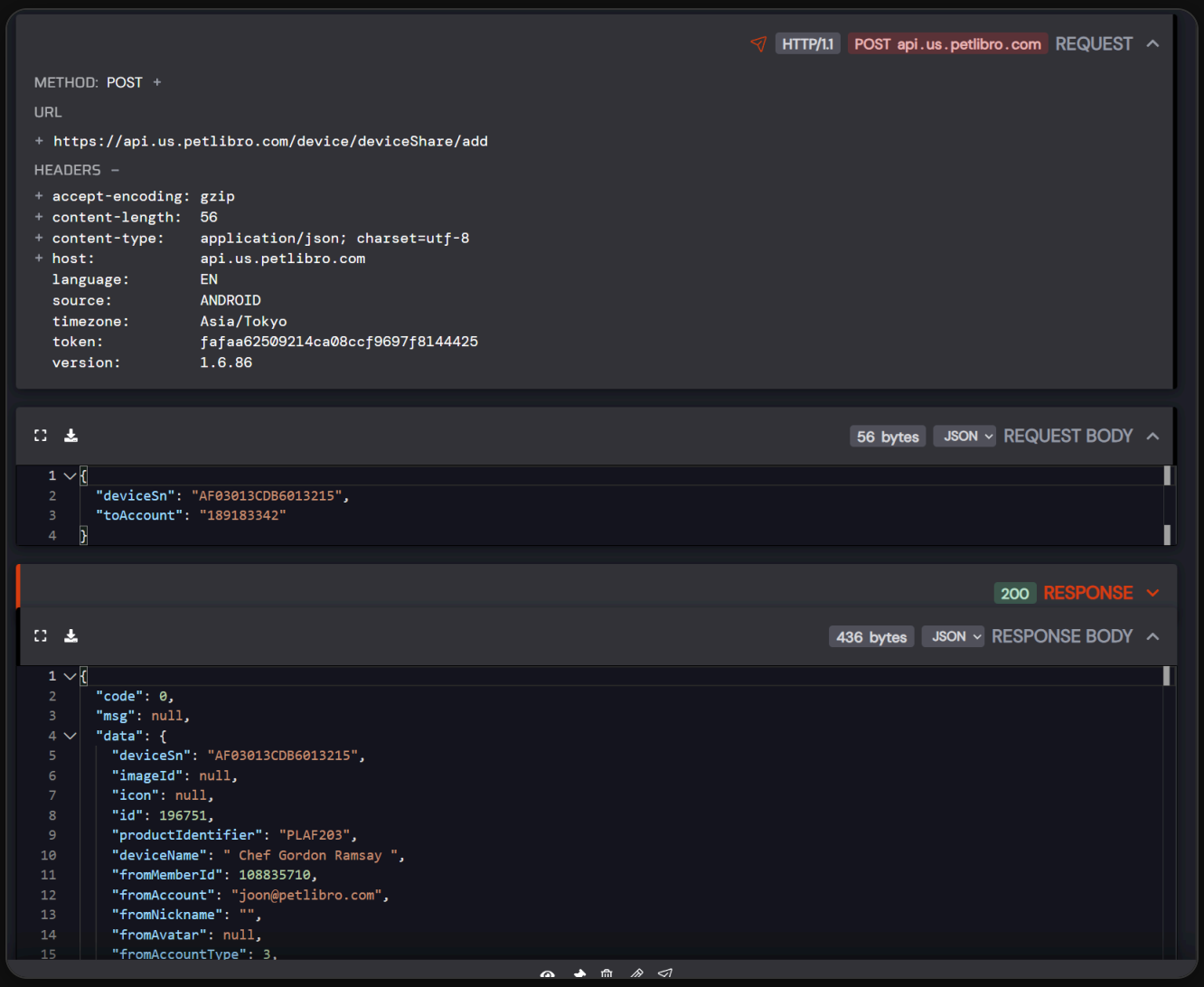
Petlibro devices let owners record mealtime messages for their pets. These audio IDs are sequential and incrementing. The `/device/deviceAudio/use` endpoint lets you assign ANY audio to ANY device:



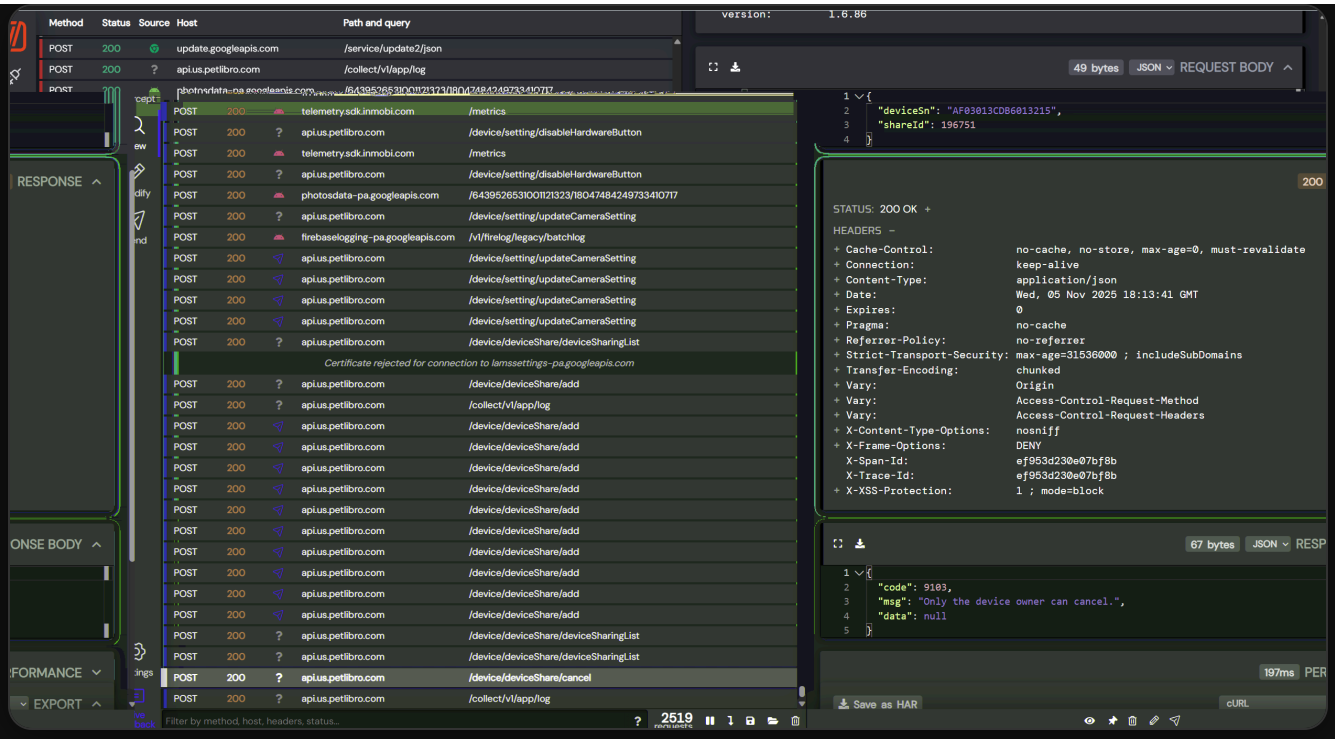
Then just fetch the device info again and you get the audio URL. I grabbed 5 random people's mealtime recordings during testing. Someone could easily automate this to harvest all recordings.

6. Add Yourself As Shared Owner

The only endpoint that checks permissions is REMOVING a shared owner. But ADDING one? Wide open.



You can add yourself as a shared owner to anyone's device. The response even shows the original owner's email (`joon@petlibro.com` - nice, an employee's device).



The Disclosure Timeline

November 5, 2025: Reported all 6 vulnerabilities with full details and screenshots.

November 6, 2025: They acknowledged receipt, said they'd assess within 5 business days.

November 8, 2025: They offered $500 bounty. Said they'd fix "priority issues next week, secondary issues before end of year, and third-party login issue early next year."

November 9, 2025: I pushed back HARD on the timeline. The auth bypass is a complete account takeover - you can't push that to "early next year." Also asked them to reconsider the bounty given the severity (industry standard would be $3-5k for this scope).

November 13, 2025: They stuck with $500, citing "startup budget constraints." Fair enough. Asked for my ACH details and said they would send right away.

November 14, 2025: Sent ACH details.

November 18, 2025: They sent a confidentiality letter AFTER I sent my payment info, asking me to sign it. I never agreed to any NDA.

November 19, 2025: They processed the $500 payment anyway, saying they "trusted me to sign it."

November 23-29, 2025: Multiple follow-ups asking me to sign the confidentiality letter. I didn't reply.

November 30, 2025: I explicitly refused to sign. Clarified that I never agreed to any NDA, the payment was sent without a signed agreement.

December 4, 2025: They confirmed "majority of issues resolved" and that the auth bypass was "fixed in the latest app version." They said they're keeping the old vulnerable endpoint active for legacy compatibility and monitoring upgrade rates, planning to enforce upgrades "within two weeks."

December 27, 2025 (Today): The old vulnerable `/member/auth/thirdLogin` endpoint is STILL ACTIVE. They made a new secure endpoint but left the old one working for "compatibility." Anyone can still login to any account. It's been over three weeks since they said "two weeks."

The NDA Situation

Let me be clear about what happened:

- I reported vulnerabilities with no mention of any NDA
- They offered $500 and asked for ACH details and said they would pay me right after no strings attached.
- I sent ACH details
- AFTER I sent payment info, they sent a confidentiality letter
- I never replied agreeing to anything
- They sent payment anyway, saying they "trusted me to sign"

- They followed up 4+ times asking me to sign
- I explicitly refused

You can't send someone money and then claim they agreed to terms they never accepted. That's not how contracts work.

Why This Matters

Petlibro is one of the biggest smart pet feeder companies. Their products are in millions of homes. These vulnerabilities allowed:

- **Complete account takeover** of any user
- **Access to private audio recordings** (people talking to their pets - intimate moments)
- **Device hijacking** - change feeding schedules, access cameras
- **Pet data exposure** - names, breeds, health info, photos

For pet owners who travel and rely on these devices to feed their animals, a malicious actor could:

- Stop feeding someone's pet
- Access their home cameras
- Harvest personal data
- Listen to private recordings

To Petlibro

You fixed most of the issues. That's good. But leaving the authentication bypass endpoint active for 2+ months because you're waiting for "upgrade analytics"? That's not acceptable.

You told me on December 4th that forced upgrades were coming "within two weeks." It's now December 28th - over three weeks later. The vulnerable endpoint still works. Every day you leave it active is another day any attacker can take over any account.

"Legacy compatibility" is not a valid reason to leave an authentication bypass live. Force the upgrade. Break old versions. Your users' security matters more than a smooth upgrade experience.

Also, trying to get researchers to sign NDAs AFTER they've already reported and AFTER you've sent payment is sketchy. Don't do that.

Your users trust you with access to their homes and their pets' wellbeing. Act like it.

Will update once they removed the old endpoint.

Update (2025-12-28)

They emailed me saying they removed the endpoint after I told them about the blogpost

---