YUCP and it's controversial use case with The Druffle (VRChat avatar)

Written: 03/08/2026 (DD/MM/YYYY)
Last Updated: 04/08/2026 (DD/MM/YYYY)

This paragraph is hereby a declaration of passivity and is not in any way, shape or form targeting the creators of the avatar described in the following document. Please do not use this for any targeted harassment, this document's purpose is only to outline the obscurities and qualms about the use of YUCP and the Druffle avatar. Any targeted harassment referencing this document is not at all indorsed by me and I have no connections or responsibility for such.

YUCP -> https://github.com/Yeusepe/YUCP-Creator-Assistant?ref=jinxxy
The Druffle Avatar -> https://jinxxy.com/LunarArray/Druffle

Details of the following document
Green -> Confirmed True with evidence
Amber -> Strongly likely that it is true, although no evidence/not enough
Red -> No evidence

Extra information will be written in *Italics*

Clarification of vocabulary and definitions
Jinxxy -> Online VR marketplace, located at https://jinxxy.com, where the Druffle is sold.
Druffle -> Avatar utilising the YUCP system, link above.
YUCP -> An avatar system that allows creators to "gate Discord access (or other benefits) for paying customers."
(https://github.com/Yeusepe/YUCP-Creator-Assistant?tab=readme-ov-file)

It has come to my attention that the creators have suffered harassment, DO NOT HARASS THEM, its just a VRChat avatar, even if it wasn't handled correctly.

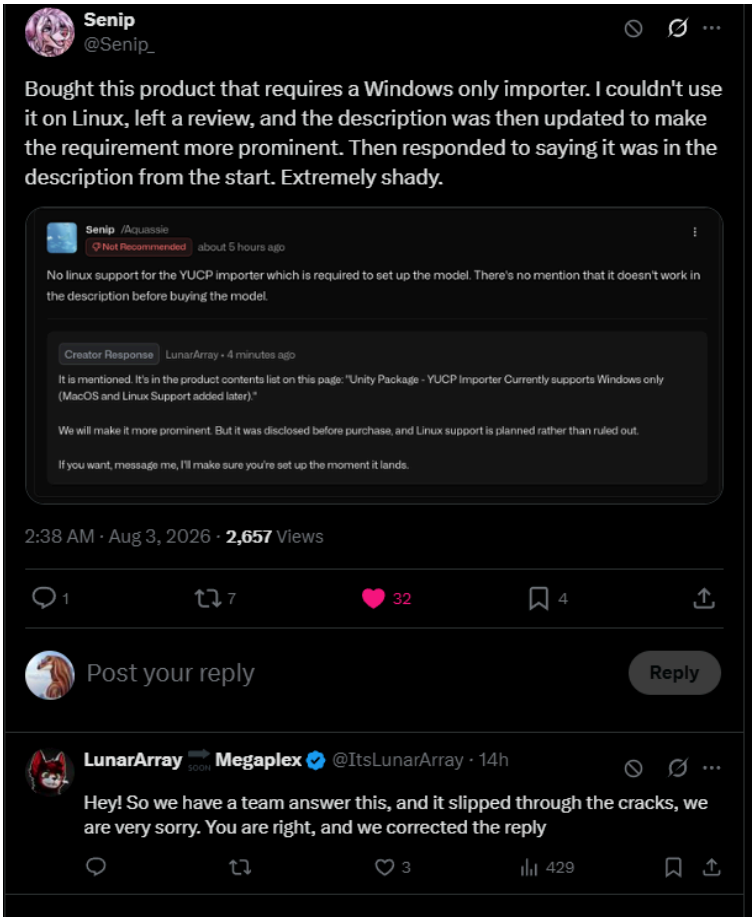
Introduction

Ok, I've been watching this for quite a while now and I believe (in my opinion) that this has been misrepresented by the creators and individuals have been misinformed about this product. So I have decided to do some digging into the product and the YUCP system.

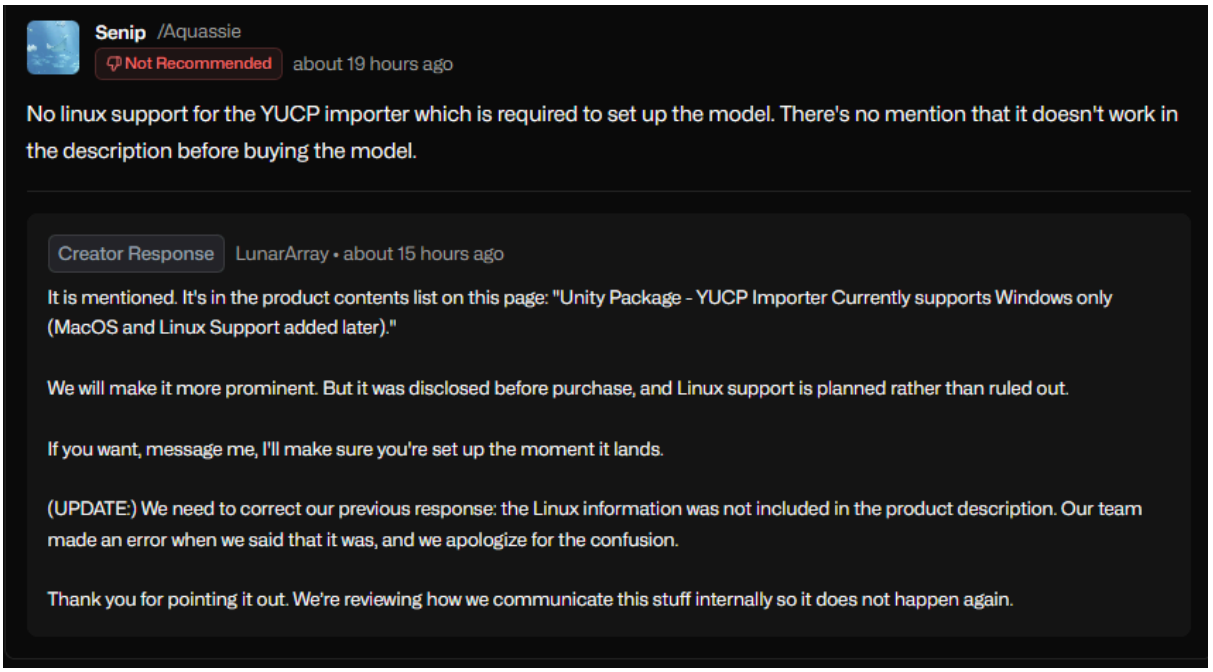
Misinformation and Misrepresentation

This system doesn't protect the base product, it's just requiring additional, unnecessary data from the customer which has already been bypassed.

Not to mention that I, as well as many others have noted that there was initially no mention of the system upon release, nor any mention of OS support. Then it appears to have been silently added in, then the initial reviews calling this out were replied to stating it is in the description.



This image shows the creator of the Druffle avatar confirming the YUCP system was missed out of the initial release. But, then within the review section mentioning this, the creator corrects them saying that it was there, stating it was disclosed before purchase, but it wasn't.

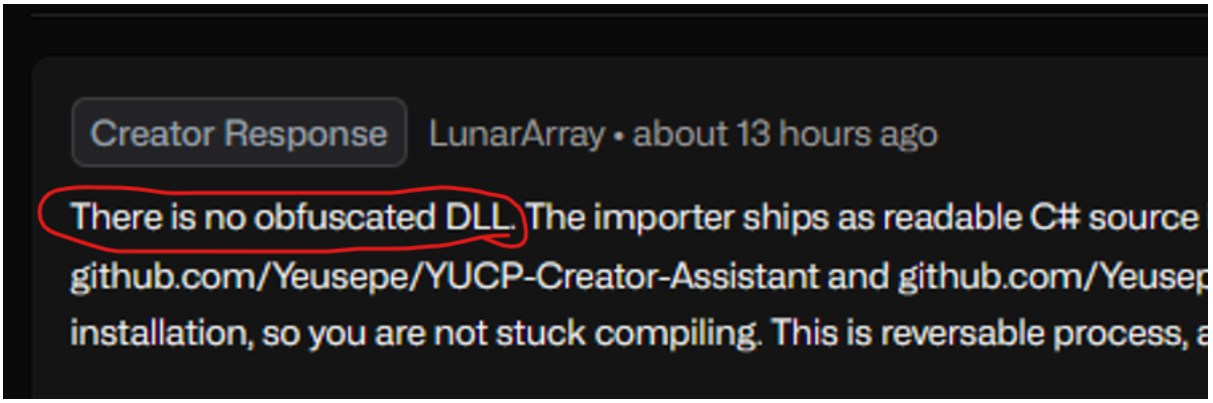


Furthermore the creator has been replying with misrepresentations of the YUCP system by definition YUCP is a DRM.
The Wikipedia definition for a DRM is as follows:

> **Digital rights management** (**DRM**) is the management of legal access to digital content. Various tools or **technological protection measures**, **such as access control technologies**, can restrict the use of proprietary hardware and copyrighted works. *https://en.wikipedia.org/wiki/Digital_rights_management*

This software actively uses access control technologies requiring the use of discord and/or licensing. By definition making it a DRM.

Furthermore the creator replied to a review stating there are no compiled binaries, but there is a DLL (which is a compiled binary) it is compiled and not human readable without specific software. (YUCP.DirectVpmInstaller.Runtime.dll)

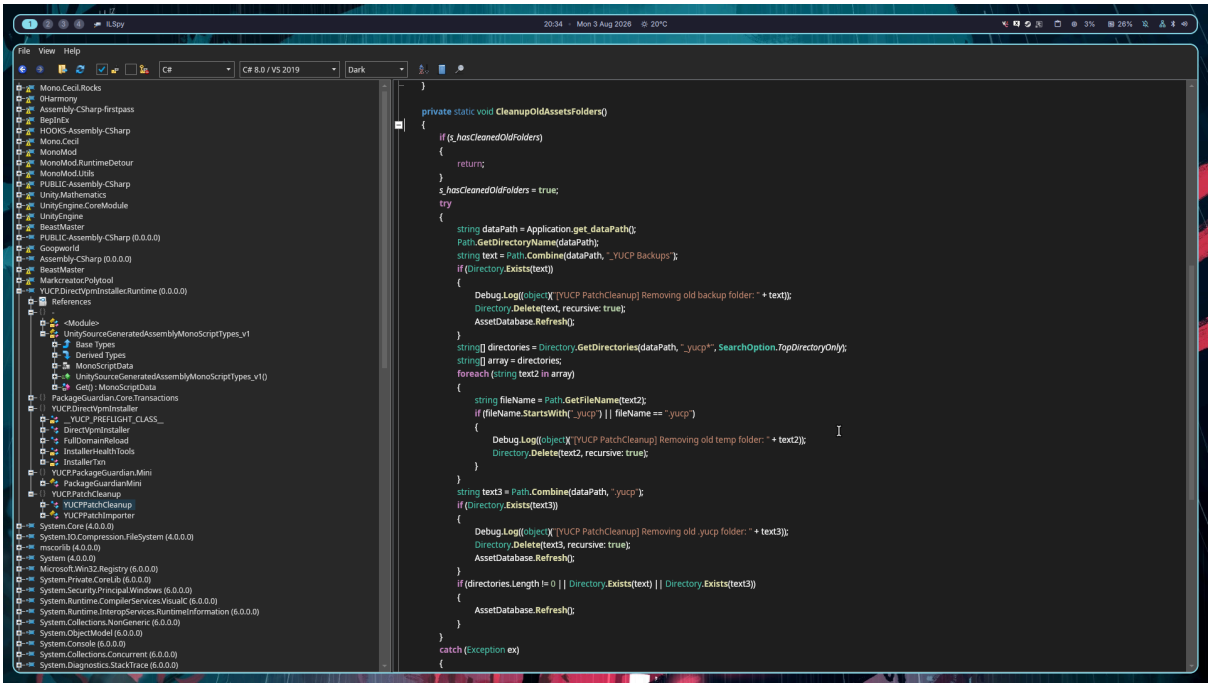


Compiled binaries are dangerous within Unity, especially when unknown to the end user. When importing a script or DLL within Unity, it will automatically run, without any input from the user apart from clicking import.

When writing a unity script, you can implement the [InitializeOnLoad] attribute to a method. This runs the code within the method as soon as imported. We can tell that this attribute belongs to a method within the script imported because the verification window appears when imported without user input. We do not know what else is being run by this script. There is currently no reason to believe that it is malicious but it is good to keep in mind because it was never mentioned before purchase.

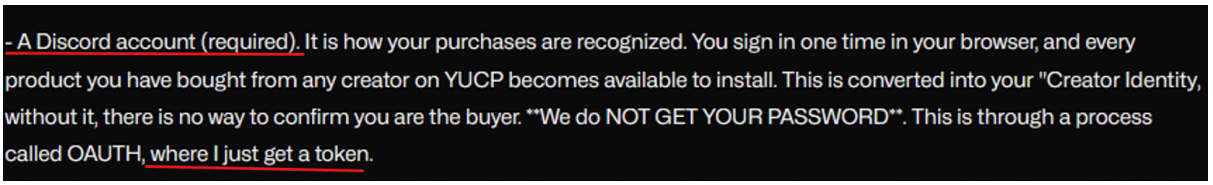
The DLL (YUCP.DirectVpmInstaller.Runtime.dll) is within the project.

The DLL is **not obfuscated.** Dedicated software is required to read it, such as dnspy. Here is the non obfuscated version:

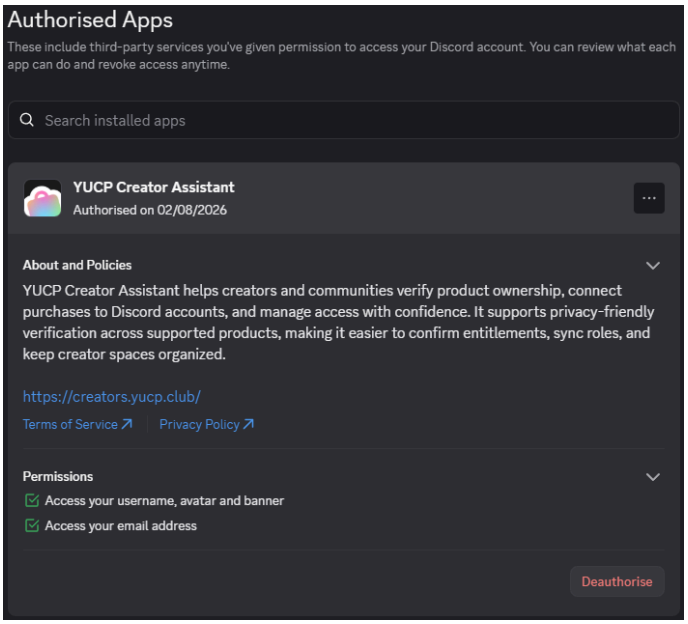


I previously incorrectly stated it was obfuscated, thank you to bracketproto for catching my silly mistake.

Furthermore, when verifying, you must link your discord account using a discord bot, and then provide a license key. They state that they only receive a token, it is not specified what this token is so it is impossible to say what information they are getting. Most likely this is just some generated token to link your account to your purchase.



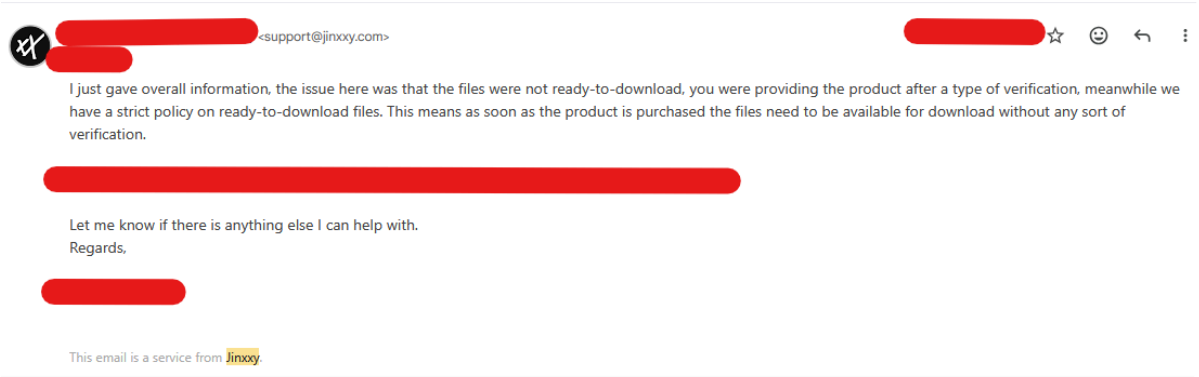
But, when we take a look at the permissions granted to the bot when you connect, it has access to your username, avatar, banner and **email address.**



The creators themselves may not receive your email address and username but there is no mention until you attempt to link your account that discord will share your email address and username. Someone, most likely the creator of YUCP, is receiving your email address and username associated with your account.

Jinxxy TOS, Package download and Download server.

When purchasing the product on Jinxxy, you do not immediately receive the product. You receive the installer which requires verification and discord connection. This is against Jinxxy's TOS.
Here is an email from the official Jinxxy support email detailing the point:



Repeated in text is the following:
"I just gave overall information, the issue here was that the files were not ready-to-download, you were providing the product after a type of verification, meanwhile we have a strict policy

on ready-to-download files. This means as soon as the product is purchased the files need to be available for download without any sort of verification. “

Furthermore the exact part of the Jinxxy Marketplace Rules and Guidance this falls under is detailed below:
https://support.jinxxy.com/hc/en-us/articles/16736843446669-Marketplace-Rules-Guidelines

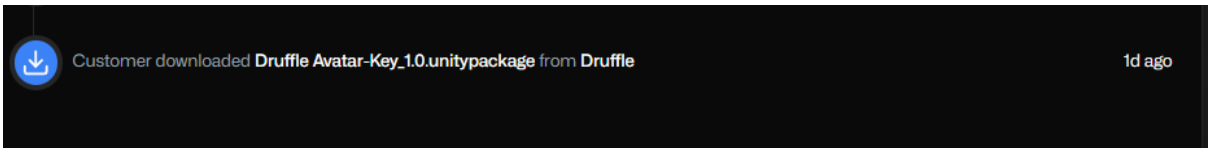
> ***Product Listings:***
>
> - *Products must be **ready-to-download files.***
> - *Listings for pre-orders, promotions, sales, commissions, or services are not allowed.*
> - *You must clearly state what is included in your product.*

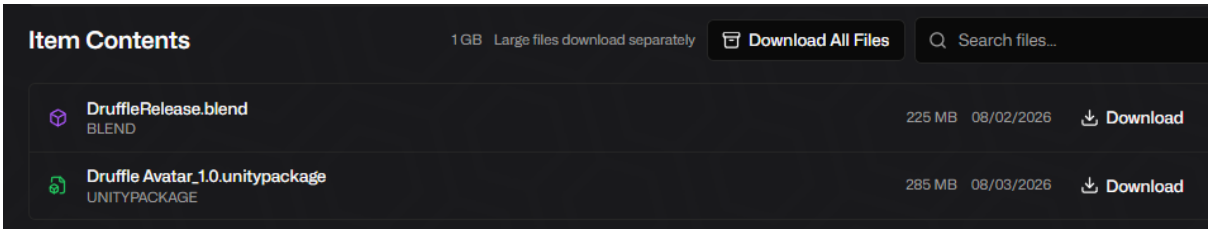
4. Product Listings:

- Products must be ready-to-download files.
- Listings for pre-orders, promotions, sales, commissions, or services are not allowed.
- You must clearly state what is included in your product.
- If your product contains DRM, licensing restrictions, anti-piracy software, or any data collection, you must:
  - Disclose these restrictions upfront.
  - State what data is collected and how it is used.
  - Collect only the minimum data needed to verify a license, and nothing more.
  - Keep the tool within its own project folder. It may not access the user's system or any other data.
  - The tool should not auto-update or change its behavior after download.
  - Provide the tool source code if we request it.

The files are downloaded from an external server after verification, I cannot provide details on the server such as hosting provider, etc, but this breaks the Jinxxy TOS.
If the server is down, you cannot receive your file.

(Update)
After being updated, it has now been changed to the full .unitypackage.
In the eyes of documentation, here is the change between the two files (pre-update, then post-update)





Fingerprinting

To understand the following, you will need some context on what Hashing is. If you already know hashing, feel free to skip the next paragraph.

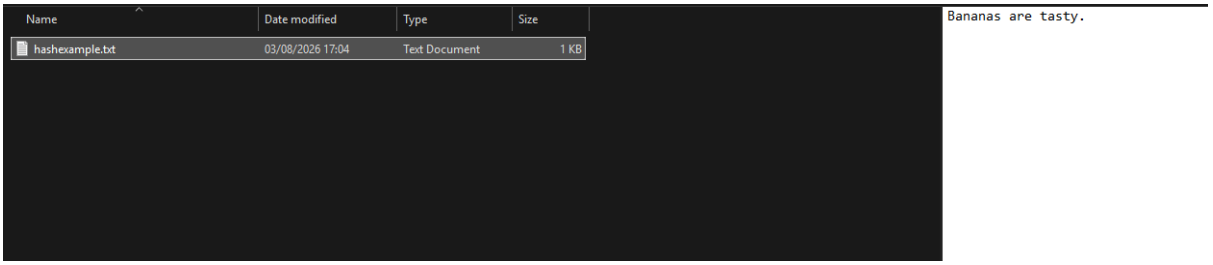
Hashing is a way of irreversibly generating a fixed length string on a file. An example of where this is used is antiviruses. Antiviruses cannot match millions of bits to a database within a reasonable time, so it generates a unique fingerprint of a file called a Hash. It is impossible to receive the original file from a hash, it is one way and used to verify file integrity, and check if files are the same. If two files are the same, and the same hashing algorithm is used on both files, they will ***always*** *produce the same hash.*

The following are the SHA256 hashes of two separate people's Druffle FBX.

42[...]95a[...] Druffle.fbx
40[...]12c[...] Druffle.fbx
To double check, another individual provided their hash in SHA256, and provided this:
5C[...]216[...]

These are hashes of the original file, no original data has been shared and thus does not break the TOS of the Druffle VRChat avatar, the hash is completely irreversible and no original data can be obtained. They have been truncated to ensure full anonymity of the providers.
As seen, they are not the same. This means that the file is not the same bit for bit, which it should be. These files were downloaded upon release and are the same version of the avatar (1.0.0).

Example:
Here is a text file (imagine its an fbx file for this demonstration)

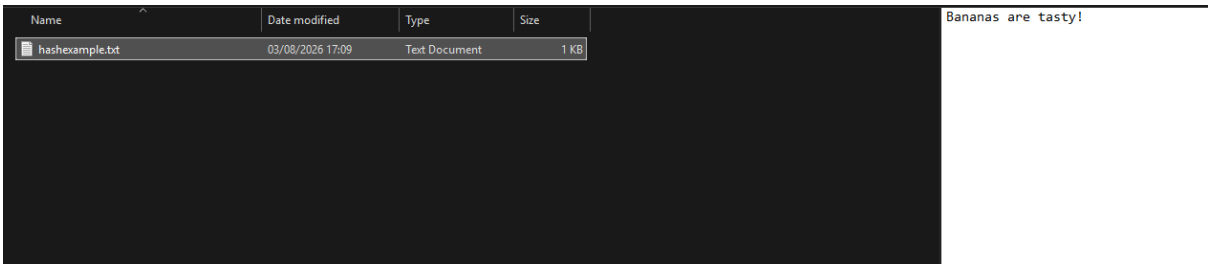


The file contains the text "Bananas are tasty."
When we generate a sha256 hash for this file using the following command:

```
Get-FileHash -Path "C:\path\to\your\file.ext" -Algorithm SHA256
```

Then we get the following hash:
087C23B32CA5B9E448C85CEC705562FF59FA7AC9BC13E363550AF6DE0346892C

But if we now change the text to have an exclamation mark at the end, now saying “Bananas are tasty!”



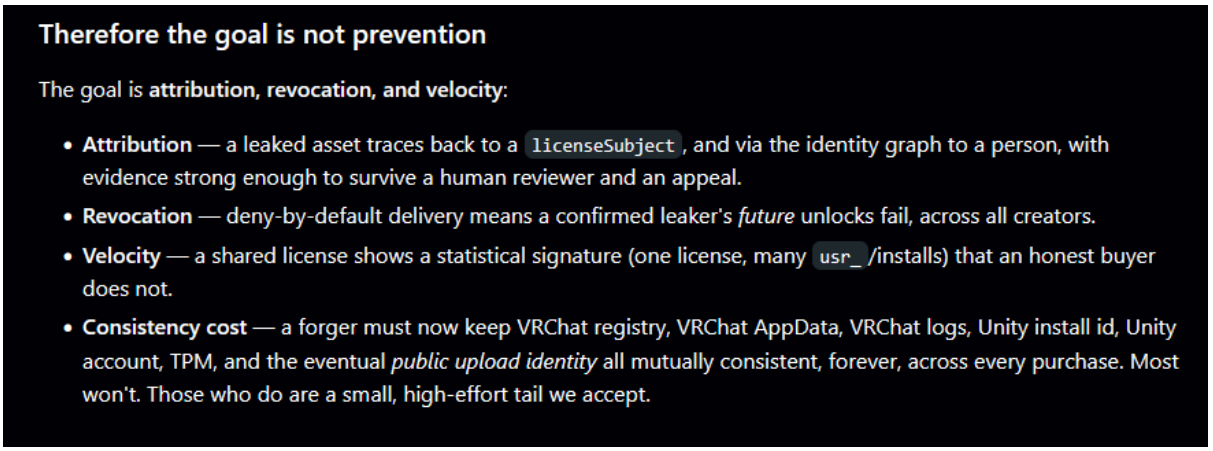
Then we get the following hash:
6A320F54F434840F8E36B0FE576F1B1938F940D8B7DA0D20F2121F54DCED28D6
As we can see, even though only one character has been changed, the entire string is significantly different, and fixed length.

This could point to some form of fingerprinting, which would mean that the file is altered per person, likely indicating who is meant to own said file, likely to prevent piracy and sharing to identify the person sharing the file.

There wouldn’t be a problem with this, but the creator has denied that it does this, and it is not mentioned anywhere within the TOS or the Avatar listing as of writing. Although from what information I have access to, the only reasonable explanation is some form of fingerprinting to uniquely identify an individual based on their unique FBX. The files are the same version and are not customised on import, so the only answer is each FBX is different.

(Updated 04/08/2026)

https://github.com/Yeusepe/YUCP-Creator-Assistant/tree/24cbd57c2cd19dae0bbe459bb3a567d2aff3f63f?tab=License-2-ov-file#therefore-the-goal-is-not-prevention



Therefore the goal is not prevention

The goal is **attribution, revocation, and velocity**:

- **Attribution** — a leaked asset traces back to a `licenseSubject`, and via the identity graph to a person, with evidence strong enough to survive a human reviewer and an appeal.
- **Revocation** — deny-by-default delivery means a confirmed leaker's *future* unlocks fail, across all creators.
- **Velocity** — a shared license shows a statistical signature (one license, many `usr_` /installs) that an honest buyer does not.
- **Consistency cost** — a forger must now keep VRChat registry, VRChat AppData, VRChat logs, Unity install id, Unity account, TPM, and the eventual *public upload identity* all mutually consistent, forever, across every purchase. Most won't. Those who do are a small, high-effort tail we accept.

YUCP does indeed fingerprint your file.

YUCP and Major Security Concerns

YUCP is the system used for the Druffle avatar. There have been many major concerns for privacy and security raised about this system.

(Updated 04/08/2026, DD/MM/YYYY)

First of all, **do not panic.** I have found **no evidence of intentional malicious behaviour.** However if you are just glancing, I would still recommend changing your password and avoid running YUCP. I could have easily missed something, but vrchat credentials are definitely handled by this program.

Are my credentials leaving my computer?
https://github.com/Yeusepe/YUCP-Creator-Assistant/blob/24cbd57c2cd19dae0bbe459bb3a567d2aff3f63f/packages/providers/src/vrchat/client.ts#L260

```
60    async beginLogin(username: string, password: string): Promise<VrchatBeginLoginResult> {
61      // Per VRChat OpenAPI spec, /auth/user has parameters: [] and security: [{authHeader: []}]
62      // meaning it uses HTTP Basic auth only, no ?apiKey= query parameter.
63      // Adding ?apiKey= to this endpoint causes VRChat to return 401 (Missing Credentials).
64      // Use request() directly to match the same contract as the working VrchatWebClient.
65      const { response, data } = await request('/auth/user', {
66        method: 'GET',
67        headers: {
68          authorization: buildBasicAuth(username, password),
69        },
70      });
```

Yes, if provided the credentials are sent from your PC, see the function linked above.
But from what I can tell, they are **only sent to the official VRChat API.**

```
const VRCHAT_API_BASE = 'https://api.vrchat.cloud/api/1';
const VRCHAT_USER_AGENT = 'YUCP Creator Assistant/0.1.0 (https://yucp.app)';
const AUTH_COOKIE = 'auth';
const TWO_FACTOR_AUTH_COOKIE = 'twoFactorAuth';
interface VrchatApiConfig {
  clientApiKey?: string;
}
```

https://github.com/Yeusepe/YUCP-Creator-Assistant/blob/24cbd57c2cd19dae0bbe459bb3a567d2aff3f63f/packages/providers/src/vrchat/client.ts#L26

From what I can tell, this is used to verify ownership of an avatar, it does not seem to have any malicious intent.

The session token is also being used here:
https://github.com/Yeusepe/YUCP-Creator-Assistant/blob/24cbd57c2cd19dae0bbe459bb3a567d2aff3f63f/convex/plugins/vrchat.ts#L33

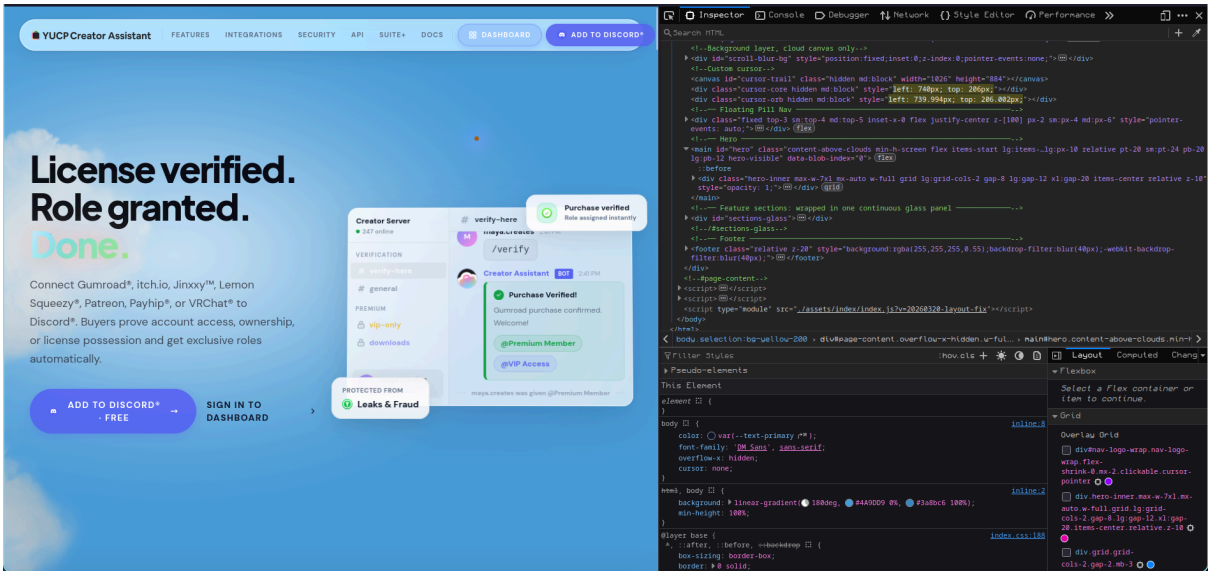
```
interface VrchatSessionTokens {
  authToken: string;
  twoFactorAuthToken?: string;
}
```

Again, there seems to be no malicious intent here.
Although, the session tokens are encrypted and stored within a database, stored in whatever backend internalAdapter and passed in via ctx.context.
https://github.com/Yeusepe/YUCP-Creator-Assistant/blob/24cbd57c2cd19dae0bbe459bb3a567d2aff3f63f/convex/plugins/vrchat.ts#L213

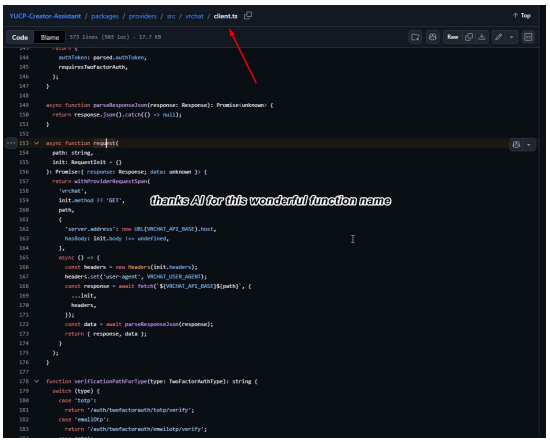
```
    let user: any;
    if (existing) {
      user = existing.user;
      await ctx.context.internalAdapter.updateAccount(existing.account.id, {
        accessToken: encryptedSession.accessToken,
        idToken: encryptedSession.idToken,
      });
    } else {
      const created = await ctx.context.internalAdapter.createOAuthUser(
        { name: displayName, email, emailVerified: false },
        {
          ...buildBetterAuthOAuthAccountIdentity('vrchat', vrchatUserId),
          accessToken: encryptedSession.accessToken,
          idToken: encryptedSession.idToken || undefined,
        }
      );
      user = created.user;
    }
```

So in conclusion I do not believe credentials were used maliciously. But they were used. I have not in depth investigated details regarding how the details are obtained. I would **recommend to anyone who used this to change their password to be safe.**

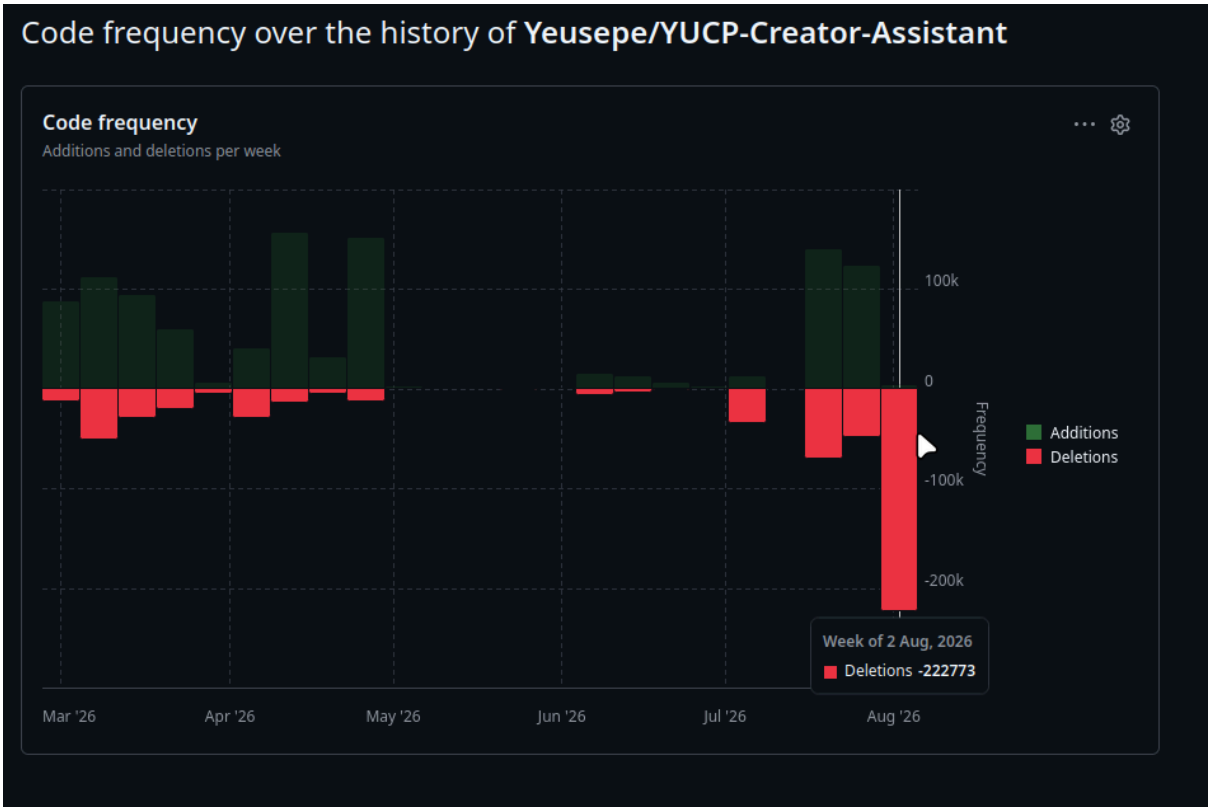
There is also a lot of evidence pointing to YUCP being vibecoded. The code shows many comments and odd naming conventions which would indicate AI generated code.



- Lots of comments
- Common AI errors such as nav bar being slightly off (see discord icon)



(Images from https://x.com/Sesilaso)



There is also a large volume of code deletions, likely related to AI generated code. But it could also be related to testing, but it's worth mentioning.

Hidden TOS

Following is section 23.1 of the Extended TOS for the Druffle (Downloaded 03/08/2026 (DD/MM/YYYY))

The following TOS provides predatory access to your machine (computer) whenever they like to “Audit” your purchase.

You can find an archive of the original TOS here:
https://coppsu.dev/YUCP&DRUFFLE/Extended%20TOS.pdf

23. Audit and Technical Monitoring

23.1 Audit Rights.

Provider may, upon reasonable prior notice and during normal business hours, audit your records and systems to verify your compliance with these Terms, including verification of license counts and permitted uses. Audits shall be conducted in a manner that reasonably minimizes disruption to your business. If an audit reveals underpayment of fees or material non compliance, you shall

---

Last Update: Jun 2, 2026 3:00 PM CST

promptly pay any underpaid amounts and reimburse Provider for reasonable audit costs.

Final Notes

I just want to clarify that after all the pushback, they have removed the DRM system. But I do not believe this excuses the creators for their misrepresentation of their system (YUCP) and their product. I do not wish any harassment towards the creators, but I as well as many others do believe (in our opinions) that the actions performed by the creators were predatory and dishonest.
I hope the creators reflect upon this situation and thrive and manage to regain any footing lost, and trust lost.
I also want to advocate for better, non predatory protections of creators work and ensure that systems like this are not welcome within the VRChat creator community.

I DO NOT BELIEVE that there was initially bad intention, and I do not endorse the apparent “death threats” and “doxxing”. It’s a vrchat avatar, chill a little. There are is NO excuse to threaten someone especially for something as trivial as an avatar.

Thank you so much from all of the following:

Accreditation

Coppsu - Writing and collation of all evidence
FizzyxFresh - Helping with information gathering, providing evidence and being a great person.
Kiroma - Helping gather information, providing evidence and noticing hash misalignment. (Very cool)
BracketProto - Helping gather information, providing evidence and a very nice person.