If you have ever visited a website protected by the Anubis anti-bot system, you might have noticed two specific images: pensive.webp and happy.webp. Maybe you wondered what these images mean and why they appear during your visit. In this post, I'll explain what those images represent, why Anubis shows them, and break down the anti-bot technology like Proof-of-Work in simple terms.

## Why Do Anti-Bot Pages Exist?

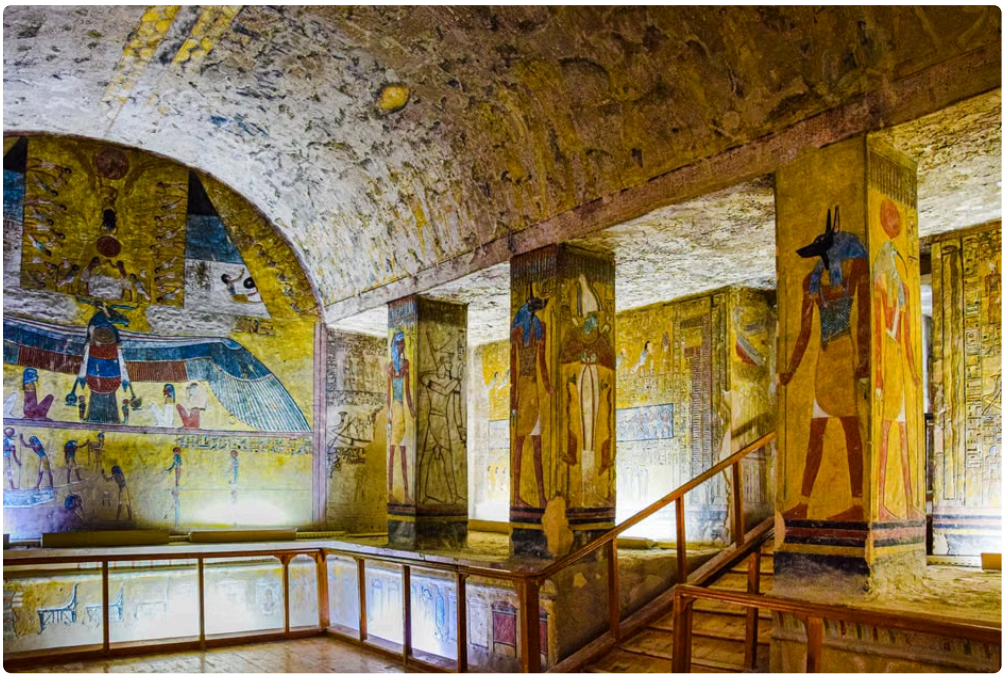
Before diving into the Anubis UI elements, let's talk about why websites show anti-bot pages in the first place.

- **Preventing Abuse:** Bots can cause serious trouble, like scraping content, spamming forms, or skewing analytics data.
- **Protecting Infrastructure:** High-volume automated traffic can overload servers and slow down or crash websites.
- **Ensuring Fair Access:** Bots often hog resources unfairly, making sites slow or inaccessible to genuine users.

To handle these challenges, many sites deploy anti-bot protections that can identify automated scripts and stop them or make them work harder before granting access.

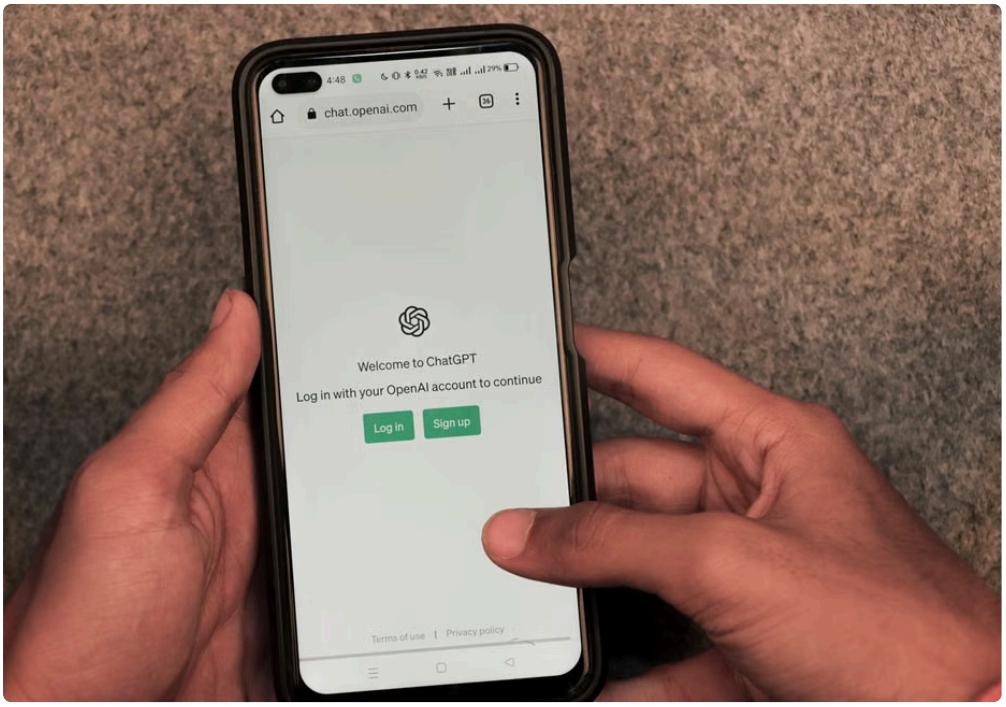
## What is Anubis and Its UI Elements?

**Anubis** is a modern anti-bot and abuse-prevention tool that websites embed to protect themselves from automated traffic. When you see the Anubis page, it typically shows two core UI elements represented by two images:



- pensive.webp: A thoughtful or "pensive" face, showing that your browser or connection is under evaluation.
- happy.webp: A smiling or "happy" face, indicating your request has passed the checks and the page is loading.

These images are not just decoration. They provide visual feedback about where you stand during the bot-detection process.




### Is it Normal to See Both Images?

Yes, it's completely normal to see both pensive.webp and happy.webp consecutively during your visit. Here's a typical flow:

1. **Initial Request:** Your browser first receives the "pensive" state, indicating Anubis is checking whether you are a bot or a real human.
2. **Proof-of-Work:** Your browser will perform some computational work behind the scenes without needing you to do anything.
3. **Verification Complete:** When the checks finish, the "happy" image appears, signaling that you're cleared and page loading can continue.

In short, seeing both images shows the progression of the bot mitigation process and is how Anubis communicates status.

## Proof-of-Work in Plain English

If you are wondering what is this "Proof-of-Work" behind the scenes, here is a simple explanation:

- **Definition:** Proof-of-Work (PoW) is a task that requires your device to use a little bit of computing power to solve a puzzle. This is easy for a real computer (like your browser) but hard or time-consuming to do at scale for a bot farm.
- **Why Use It?:** By requiring this mild, automatic effort, websites can slow down automated abuse attempts without bothering genuine users with challenges like CAPTCHAs.
- **How It Works:** When you visit the site, your browser solves a small math puzzle and returns the answer. If it's correct, you move on (happy.webp). If not, the check may repeat (pensive.webp).

This way, the system confirms the visitor is a real browser doing real work, without extra clicks or puzzles.

### Real-World Analogy:

Think of it like a turnstile that requires you to push gently to enter. A person can do it easily once. But if hundreds of fake visitors want to push millions of times, it slows them down enough to keep it manageable.

## Background: Hashcash and Other Proof-of-Work Systems

Proof-of-Work is not new—one of the earliest widely known PoW systems is called *Hashcash*, invented to fight email spam.

Feature Description Invented 1997 by Adam Back Purpose Reduce spam emails by requiring senders to compute a small cryptographic hash before sending Method Computes a partial hash with set difficulty — easy for regular use, hard to mass-produce Modern Applications Anti-spam, anti-DDoS, cryptocurrency mining (e.g., Bitcoin)

Websites like those using Anubis borrow this notion to require visitors' browsers to spend tiny amounts of CPU time solving similar puzzles. This discourages bots that seek to send large numbers of bad requests quickly.

## Why JavaScript and Modern Browser Features Matter

Anubis relies heavily on the capabilities of your browser to run complex JavaScript and perform the Proof-of-Work calculation behind the scenes.

- **JavaScript Execution:** Your browser must be able to run JavaScript without restrictions. Blocking or disabling JS can stop these checks from working.
- **Modern Features:** Use of modern web features such as WebAssembly and performant Crypto APIs allows faster and more efficient puzzle solving.
- **Compatibility:** Browsers like Chrome, Firefox, Safari, and Edge handle these scripts well, but older or limited browsers may get stuck showing the pensive image or time out.

If Anubis cannot run its scripts correctly, you might see boerse-social the "pensive.webp" image stuck on screen or repeated multiple times. This is a signal that the page cannot verify your legitimacy automatically.

### Checklist to Avoid Issues with Anubis Pages

1. Make sure JavaScript is enabled in your browser.
2. Use a modern browser version (updated Chrome, Firefox, Safari, Edge).
3. Disable browser extensions that block scripts or perform aggressive ad-blocking.
4. Ensure your connection isn't being routed through suspicious proxies or VPNs that might look automated.

## Summary: What Do pensive.webp and happy.webp Mean for You?

- **pensive.webp** means Anubis is working to check if your visit is legitimate.
- **happy.webp** means you've passed the test and can proceed to the site.
- This process is powered by Proof-of-Work puzzles running in modern browsers using JavaScript.
- It's normal and expected to see both images during this short verification phase.
- If stuck on the pensive image for a long time, verify your browser supports JavaScript and isn't blocking scripts.

## Final Thoughts

In web operations and abuse prevention, transparency with users is important. Anubis's use of clear visual feedback such as pensive.webp and happy.webp is a thoughtful way to let you know what's happening behind the scenes.

So next time you encounter those images while waiting to enter a website, remember: your browser is quietly proving it's real, one small math puzzle at a time.

If you're a site operator curious about adding or tuning such systems, start by making sure your pages serve lightweight checks and explain flows simply to site visitors — it makes everyone's experience smoother.