

```html

If you've ever visited a website running anti-bot protections like Anubis Proof-of-Work, you might have noticed the page seems to "pause" for a moment or that your laptop fans suddenly kick in. You're not imagining it: these anti-bot pages can indeed feel slow or heavy on your browser's CPU. So why does this happen? Is this normal? And what exactly is Proof-of-Work anyway?

In this blog post, we'll break down in simple terms:

- Why anti-bot pages with Proof-of-Work challenges exist
- What Proof-of-Work means and how it works, explained with minimal jargon
- A quick history lesson on Hashcash, the inspiration behind this technique
- Why your browser's JavaScript and modern CPU features matter here

## Why Do Anti-Bot Pages Exist?

Imagine you run a popular news or finance website. Hundreds or thousands of people visit regularly, but sometimes, automated programs – aka bots – try to do things they shouldn't:

- Scrape content excessively
- Attempt brute-force logins
- Launch denial-of-service attacks
- Spam forms or comments

To protect your site, you want to filter out these bots while letting real humans pass through smoothly. That's where anti-bot pages come in — they act as gatekeepers.

These pages require visitors to perform some task that a normal browser and human user can do but is costly or impossible for most bots. If the task is "hard" enough, automated tools give up, but real users only face a minor, usually invisible, delay.

## Proof-of-Work in Plain English: What Is It and Why Use It?

Proof-of-Work (PoW) is one way an anti-bot page can prove you're not a bot. The key idea:

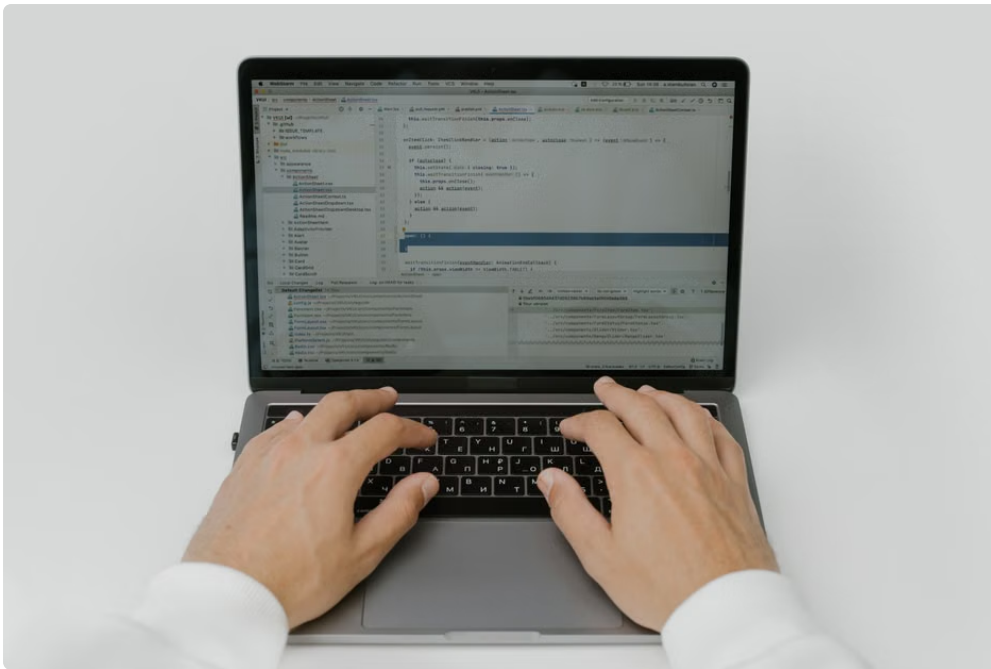
- Your browser must solve a small computational puzzle before continuing.
- This puzzle takes some effort (CPU time) but is easy for your device to verify afterward.
- Because bots want to spam or scrape *\*fast\** and at *\*scale\**, PoW slows them down significantly.

Think of it like a quick mental math quiz at a subway turnstile. Everyone can do it, but if you had to do thousands of quizzes simultaneously and repeatedly, you'd get tired fast — exactly the point for bots.

### The Puzzle: A Bit More Detail

Under the hood, the puzzle typically involves finding a number (called a "nonce") that, when combined with other data and run through a cryptographic hash function, produces a hash value with certain properties — like a specific number of zeroes at the start.

Because hash functions are one-way and unpredictable, the only way to find [is this a captcha](#) that nonce is by trying many different numbers, a process called “brute forcing.” Your CPU cycles get spent trying lots of candidates until it finds a good one.



## Hashcash: The Forefather of Proof-of-Work

Proof-of-Work is not new technology. One of the earliest PoW mechanisms was Hashcash, invented in 1997 by Adam Back.

Hashcash was designed to combat email spam. The idea:

- Before sending an email, the sender computes a small PoW puzzle.
- This “cost” is tiny for someone sending a few emails but becomes expensive for spammers sending millions.

Hashcash works by requiring a SHA-1 hash output with some leading zeros, just like the puzzles used today in various anti-bot controls.

## Why Hashcash Still Matters Today

Many modern browser-based Proof-of-Work schemes, including Anubis Proof-of-Work, utilize the same concept — a fast hash function combined with a nonce search — to prove your browser did some “work” before accessing a protected page.

## Why Is Anubis Proof-of-Work Slow on My Laptop?

Now, to the nitty-gritty: why might you experience **anubis performance** as slow or see a spike in CPU usage during the browser challenge?

### 1. Your Laptop Has to Do Real Work!

Unlike CAPTCHA where you just click boxes or solve visual puzzles, PoW requires your CPU to cycle through many hash computations. This isn’t a trivial task — it’s deliberately designed to be CPU-intensive.

So if your laptop is a few years old or has a low-power processor, the challenge can take noticeably longer, causing what feels like a slow or stuck page.

## 2. Browser JavaScript and Modern Features Are Required

All these computations happen inside your browser via JavaScript. Modern PoW relies on Web Crypto APIs and optimized JS engines, so if your browser:

- Is outdated
- Has JavaScript disabled or limited
- Runs on unusual or energy-saving power modes

...the performance will degrade, making the Proof-of-Work take more time and CPU.

## 3. Running Multiple Tabs or Background Processes

If you have many browser tabs open or other programs running, your CPU's capacity to quickly finish the puzzle decreases — making the page load slower.

## 4. The Difficulty Level

Websites can adjust how "hard" the puzzle is, a setting called difficulty. If the difficulty is set higher to weed out stronger botnets, then even fast machines will take a bit longer.

## Summary Table of Factors Affecting Proof-of-Work Performance

| Factor                               | Effect on PoW Speed/Performance                        | How to Improve                           |
|--------------------------------------|--------------------------------------------------------|------------------------------------------|
| Processor Speed                      | Slower CPUs take more time to find nonce               | Use a more powerful device (if possible) |
| Browser Version & JavaScript Support | Older browsers or disabled JS slow down or prevent PoW | Update browser; enable JavaScript fully  |
| Background Load                      | Other apps/tabs reduce CPU availability                | Close unnecessary apps/tabs              |
| PoW Difficulty Setting               | Higher difficulty = more work/time                     | None (site controls this)                |

## Why Anubis Uses Browser-Based Proof-of-Work

Because PoW runs completely in your browser, it's:

- **Private:** no user data is sent during the challenge except the proof itself
- **Flexible:** it works regardless of IP address or cookies, so it resists proxy or VPN abuse
- **Scalable:** minimal server resources are used because servers only verify the proof, not generate work

All these benefits help website operators keep sites running smoothly without intrusive or annoying challenges.

## My Proof-of-Work Is Slow — What Can I Do?

If you feel your browser CPU usage spikes and the page pauses too long during an Anubis Proof-of-Work challenge, here's a short checklist:

1. **Update your browser** to the latest stable version to ensure optimized JavaScript and crypto APIs.
2. **Enable JavaScript fully** and avoid extensions that block scripts.
3. **Close heavy background processes or tabs** consuming CPU.
4. **Use a laptop plugged into power** — power-saving modes might throttle CPU speed.

5. **Try a different device** if you consistently see very slow performances.

Beware of sites or advice to bypass protections — these challenges exist to keep websites secure and reliable for everyone.

## Wrapping Up

So yes — it is expected that Anubis Proof-of-Work feels slow and causes noticeable CPU usage on your laptop. This slowdown is actually how the anti-bot mechanism defends the websites you visit from bots and attackers. It's a tradeoff: a quick computational puzzle protects content and services without annoying users with puzzles or CAPTCHAs.

By understanding the basics of Proof-of-Work, Hashcash, and browser requirements, you can appreciate why these pages behave the way they do, and take simple steps to make your experience smoother.

If you're a site operator worried about performance or user frustration, remember that difficulty tuning and user communication go a long way toward balancing protection and usability.

Thanks for reading! Stay safe, surf smart, and keep your browsers updated.



...