

What Is the BabaDeda Loader? Analysis of a New ClickFix Malware Campaign

 morphisec.com/blog/what-is-the-babadeda-loader-analysis-of-a-new-clickfix-malware-campaign

Shmuel Uzan

June 16, 2026



During April 2026, Morphisec prevented multiple intrusion attempts against education and financial organizations and uncovered a significantly evolved version of the **BabaDeda** loader family. Earlier BabaDeda activity was known for concealing malicious payloads inside legitimate looking installer packages. This new framework keeps that same code genome but expands it into a far more capable loader built for stealth, evasion, and payload flexibility.

The attack begins with ClickFix, a social engineering technique that convinces users to run attacker supplied commands through trusted operating system utilities. From there it transitions into a multi stage loader that chains hidden PowerShell, in memory shellcode, DLL sideloading, external payload storage, and callback based execution to deliver information stealers and remote access trojans.

What makes this evolution dangerous is not any single technique. It is the way they are combined. The visible application package looks legitimate while the real payload sits inside an external storage file and is decoded only seconds before it executes. That design is built to defeat tools that rely on files, signatures, and reputation. The answer is not detecting the attack faster. It is preventing it from executing in the first place.

BabaDeda Is Back, and Built for Stealth

Through code genome analysis, Morphisec researchers tied this activity to the BabaDeda infrastructure [previously documented by our team](#). The lineage is not a guess. The loader carries an internal task handler that kicks off a workflow named BABADEDADA and embeds the constant 0xBABADEDADA in its code. The activity was seen across multiple campaigns, which means the framework is being actively deployed in real operations, not sitting in a lab.

The reborn version expands the original in three ways that matter to defenders:

- **Modular staging:** delivery, storage, execution, and payload deployment are split into separate components that each look benign on their own.
- **External payload storage:** the real payload lives inside an ordinary looking file such as List.Control.dat, kept away from the most visible executable.
- **In memory execution:** Donut and pe2shc style loaders map and run code directly in memory rather than dropping a clean executable to disk.

Inside the Attack Chain

The infection starts when a victim hits an attacker controlled site and gets a fake verification or CAPTCHA style prompt that tells them to run a PowerShell command. That command pulls a script and runs it directly in memory, which quietly launches a hidden PowerShell process and stages the next components. The loader checks the victim region and quits on Russian or Belarusian systems, then pulls follow on payloads over plain HTTP and injects them into a trusted Windows process such as svchost.exe.

One branch delivers a full .NET backdoor and information stealer that profiles the host, opens an encrypted channel to a command and control server, and can harvest browser cookies, saved credentials, and files on demand. Another branch downloads a package disguised as a normal application, called linguist.zip, that contains roughly 30 files arranged to look like real software.

That package is where the chain truly hides. A legitimate executable is used to sideload a malicious DLL, which reads the payload out of an external storage file, decodes only the bytes it needs with a simple XOR transform, and hands execution to a second DLL. That DLL abuses a legitimate Windows callback function to run the shellcode in memory. The downstream families in this set include DanaBot and the SectopRAT stealer, along with additional staging components.

Why Detection-First Security Cannot Keep Up

- **The payload lives in storage:** malicious bytes sit inside ordinary looking files and are decoded only at runtime, so disk and signature scans find nothing useful.
- **Execution happens in memory:** reflective and shellcode loaders leave little on disk for file based tools to analyze.
- **Functionality is split apart:** each file plays a narrow role, so no single component looks malicious in isolation.

- **It is built to dodge analysis:** locale checks, debugger checks, security product enumeration, and noisy fake control flow slow down both automated and AI assisted triage.
- **Detection arrives too late:** even when something is flagged, the loader has often already executed and the stealer or RAT is in place.

Prevention Before Execution: The Morphisec Approach

To stop a loader built to hide, you need a [prevention-first security model](#) that blocks malicious code before it ever runs. [Automated Moving Target Defense \(AMTD\)](#) morphs the runtime memory environment so evasive payloads cannot find their target and are blocked deterministically, before damage occurs. The Morphisec Anti-Ransomware Assurance Suite pairs AMTD with deception and adaptive AI driven defense to disrupt the chain before a stealer, RAT, or ransomware payload can establish a foothold. That is how Morphisec prevented these BabaDeda campaigns.

Prevention Beats Detection Every Time

Attackers will keep changing filenames, exports, packaging, and payloads to slip past tools that hunt for known indicators. Prevention works differently. It targets the execution techniques the chain depends on, which stays effective even as the variants change. When the payload never touches disk in a usable form and never behaves the same way twice, detection is already a step behind. Stopping execution is the only reliable answer.

Want the full technical breakdown, including the complete reverse engineering analysis and IOCs? Read the deep technical report attached to this post, then see how prevention-first defense holds up against loaders like BabaDeda.



About the author



Shmuel Uzan is a skilled security researcher with nearly a decade of experience. Shmuel holds a Bachelor of Software Engineering from Shamoon College of Engineering.

Unlock Your ROI Potential with Morphisec

[Calculate ROI](#)

Stay up-to-date

Get the latest resources, news, and threat research to your inbox.