There’s A New Stealer Variant In Town, And It’s Using Electron

By AirFleet - David

Published: 2023-08-10 · Archived: 2026-04-06 02:58:37 UTC

Our threat research team recently uncovered new npm packages that are used to download a new info-stealer variant that uses the popular Electron framework to disguise itself as a legitimate application. In this blog post, we’ll analyze the attack flow of this new info-stealer we detected and explain how it can stay undetected by abusing trusted development tools like Electron.

By building the malware with Electron, the attackers are able to inject their malicious code into a program that appears harmless. But behind its innocent look, this info-stealer’s main objective is to collect sensitive data from the victim’s machine. We’ll discuss how it operates, its capabilities, and how users can stay safe.

What is Electron and why it’s being used in malware

Electron is an open-source framework developed by GitHub for building cross-platform desktop applications using web technologies like JavaScript, HTML and CSS. It uses Chromium and Node.js under the hood. The Chromium engine is used to display web content and Node.js runs the backend code.

The nature of this framework allows attackers to spread their malicious code to multi-platforms with an ease of a flag and reduce the time it takes to develop code that will suit each system.

The Framework’s interface makes it easy to create attractive looking apps that users may be tricked into installing and can easily disguise themselves as legitimate tools while accessing powerful OS functions in the background.

Initial access

On Thursday, August 3rd, our supply chain tool alerted us with a suspicious package named **Crazydown** which runs an obfuscated script using a postinstall hook. The attacker used two techniques to try and hide their malicious intent. The first one involved a creative way to hide its actions in the package.json file using escaped Unicode characters. The second technique was to obfuscate the main file index.js using rename obfuscation and encryption to try and stay undetected.


There’s a New Stealer Variant in Town, and It’s Using Electron to Stay Fully Undetected - crazydown 1

Figure 1**. Escaped Unicode characters in the package.json file**


There’s a New Stealer Variant in Town, and It’s Using Electron to Stay Fully Undetected - crazydown 2

Figure 2. **Heavy obfuscated and encrypted index.js file**

To understand what's going on in this file, we debugged it to see how it decrypts its original strings during runtime using bitwise operations. After a couple of iterations, we found the url that downloads the second stage of the attack.


There's a New Stealer Variant in Town, and It's Using Electron to Stay Fully Undetected - crazydown 3
Figure 3. **The host server is revealed through the debugging process**

It's worth mentioning that a couple of days later, we found another 4 packages that use the same techniques to spread that malware, but this time in one version, the attacker exposed its full source code under this heavy obfuscation which you can see here.


There's a New Stealer Variant in Town, and It's Using Electron to Stay Fully Undetected - crazydown code
Figure 4. **Fully deobfuscated Index.js file**

So the main goal behind this code is to download an exe file from a discord server and execute it.

Second stage analysis

Uploading our sample to Virustotal shows us that it is fully undetected by any vendor, as the analysis gives us zero results.

Looking at the details tab, we can see that this sample is actually a Nullsoft installer which is a tool for creating Windows installers, and it's widely used due to its power and flexibility. Moreover, Nullsoft installer is an archive which means we can extract its content using tools like 7-zip.


There's a New Stealer Variant in Town, and It's Using Electron to Stay Fully Undetected - crazydown 4
Figure 5. **Virustotal results show this sample is fully undetected**


There's a New Stealer Variant in Town, and It's Using Electron to Stay Fully Undetected - crazydown 5
Figure 6. **Nullsoft installer evidence**

Upon extracting the sample, we found a bigger exe file (~134Mb) named **Fewer.exe**.

Analyzing this file in Detect It Easy tool we can see that it is actually an electron desktop application.


There's a New Stealer Variant in Town, and It's Using Electron to Stay Fully Undetected - crazydown 6
Figure 7. **A new bigger exe file was found**


There's a New Stealer Variant in Town, and It's Using Electron to Stay Fully Undetected - crazydown 7
Figure 8. **Detect It Easy results point to the Electron app**

To get the source code that was built into this electron app, we can look inside the resources folder. There we can find an app.asar file, which can be extracted using an asar plugin for 7-zip, and here we have the full source code of that sample.


There's a New Stealer Variant in Town, and It's Using Electron to Stay Fully Undetected - crazydown 8
Figure 9. **App.asar file that includes the source code of the app**


There's a New Stealer Variant in Town, and It's Using Electron to Stay Fully Undetected - crazydown 9
Figure 10. **Extracted app source code**

When looking at the batch file we can see the instruction that is being used to build this app using the electron-builder command for Windows.


There's a New Stealer Variant in Town, and It's Using Electron to Stay Fully Undetected - crazydown 10
Figure 11. **Instructions from batch file**

Analyzing the gavy.js file we found that this file is in charge of decrypting the full source code of the stealer. Replacing the final function statement with a console.log method will reveal the code.


There's a New Stealer Variant in Town, and It's Using Electron to Stay Fully Undetected - crazydown 11
Figure 12. **Original gavy.js file**


There's a New Stealer Variant in Town, and It's Using Electron to Stay Fully Undetected - crazydown 12
Figure 13. **Altered gavy.js file to decrypt its content**


There's a New Stealer Variant in Town, and It's Using Electron to Stay Fully Undetected - crazydown 13
Figure 14. **A snippet from the info stealer source code**

Evidence in the wild

While looking for evidence of this new variant in the wild, we found a github repository that seemed to be the source of this stealer and includes the name we found inside.

Inspecting closer, we can confirm that this is the source as it includes the same files we extracted from the Nullsoft installer and the same batch file that is in charge of the electron app build. This repository was created a month ago, corresponding to our analysis from Virustotal that showed the first analysis of this variant was a couple of weeks ago, and was probably initiated by the attacker to check the detection for their malware.

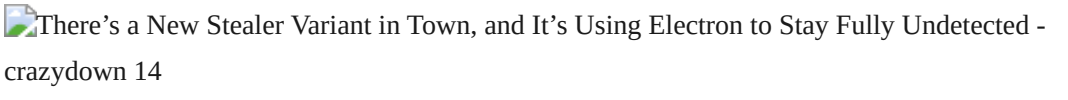
There's a New Stealer Variant in Town, and It's Using Electron to Stay Fully Undetected - crazydown 14

Figure 15. **Freshly created repository to host the source code**

Core functionalities

The core functionalities of this stealer are pretty similar to different stealers and variants of the well known “Wasp Stealer” which we detected all over the last year, both on NPM and PYPI.

It has the ability to steal cookies, browser history, wallet addresses, autofills, discord tokens, Instagram and Tiktok sessions, and much more. The main difference is that it has an auto obfuscating process using crypter.js file, and the way it is getting built using electron to look like a legitimate app and stay fully undetectable.

Conclusion

The new Electron-based infostealer depicts the constant evolution of supply chain threats. While its capabilities are concerning, the bigger issue is how adversaries are able to masquerade their tools as legitimate software.

By using trusted frameworks like Electron, they make detection way more difficult. However, through ongoing threat research and analysis, we can unravel their techniques and better protect users.

We at Mend.io are committed to this mission and will keep developing solutions to detect and block every new technique that will be used by attackers to compromise our users.
