# Widespread malware campaign seeks to silently inject ads into search results, affects multiple browsers | Microsoft Security Blog

By Microsoft Threat Intelligence

Published: 2020-12-10 · Archived: 2026-04-02 11:00:36 UTC

A persistent malware campaign has been actively distributing an evolved browser modifier malware at scale since at least May 2020. At its peak in August, the threat was observed on over 30,000 devices every day. The malware is designed to inject ads into search engine results pages. The threat affects multiple browsers—Microsoft Edge, Google Chrome, Yandex Browser, and Mozilla Firefox—exposing the attackers' intent to reach as many Internet users as possible.

We call this family of browser modifiers Adrozek. If not detected and blocked, Adrozek adds browser extensions, modifies a specific DLL per target browser, and changes browser settings to insert additional, unauthorized ads into web pages, often on top of legitimate ads from search engines. The intended effect is for users, searching for certain keywords, to inadvertently click on these malware-inserted ads, which lead to affiliated pages. The attackers earn through affiliate advertising programs, which pay by amount of traffic referred to sponsored affiliated pages.

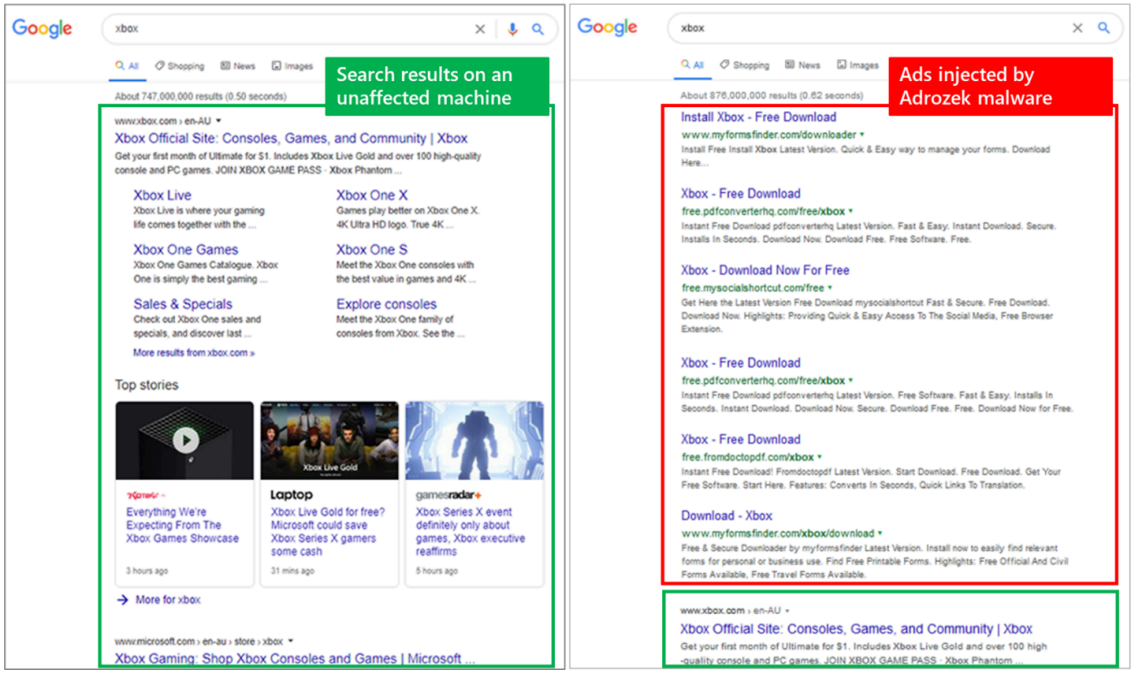


*Figure 1. Comparison of search results pages on an affected machine and one with Adrozek running.*

Cybercriminals abusing affiliate programs is not new—browser modifiers are some of the oldest types of threats. However, the fact that this campaign utilizes a piece of malware that affects multiple browsers is an indication of how this threat type continues to be increasingly sophisticated. In addition, the malware maintains persistence and exfiltrates website credentials, exposing affected devices to additional risks.

Such a sustained, far-reaching campaign requires an expansive, dynamic attacker infrastructure. We tracked 159 unique domains, each hosting an average of 17,300 unique URLs, which in turn host more than 15,300 unique, polymorphic malware samples on average. In total, from May to September 2020, we recorded hundreds of thousands of encounters of the Adrozek malware across the globe, with heavy concentration in Europe and in South Asia and Southeast Asia. As this campaign is ongoing, this infrastructure is bound to expand even further.

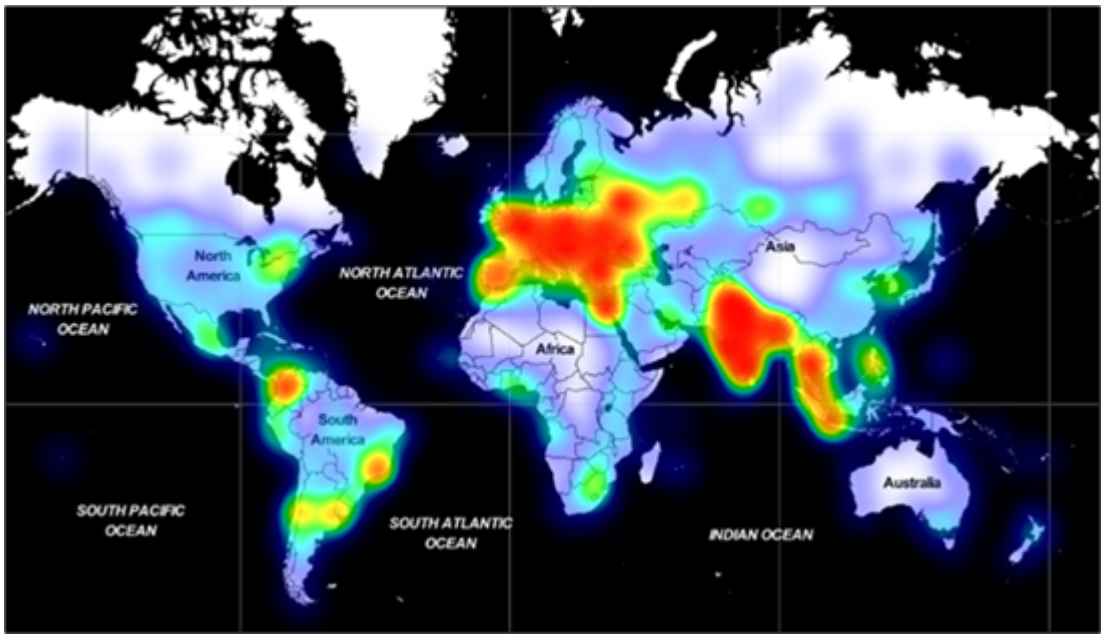


*Figure 2. Geographic distribution of Adrozek encounters from May to September 2020.*

Effectively protecting against rampant, persistent campaigns like this that incorporate multiple components, polymorphism, and evolved malware behavior requires advanced, behavior-based detection and visibility across the whole attack chain rather than specific components. Microsoft Defender Antivirus, the built-in endpoint protection solution on Windows 10, blocks this threat using behavior-based, machine learning-powered protections. For enterprises, Microsoft 365 Defender provides deep visibility into malicious behaviors.  In this blog, we'll share our in-depth analysis of this campaign, including the distribution architecture and malware behavior, and provide recommended defenses.

## Distribution infrastructure

The Adrozek malware is installed on devices through drive-by download. In our tracking of the Adrozek campaign from May to September 2020, we saw 159 unique domains used to distribute hundreds of thousands of unique malware samples. Attackers relied heavily on polymorphism, which allows attackers to churn huge volumes of samples as well as to evade detection.

While many of the domains hosted tens of thousands of URLs, a few had more than 100,000 unique URLs, with one hosting almost 250,000. This massive infrastructure reflects how determined the attackers are to keep this campaign operational.

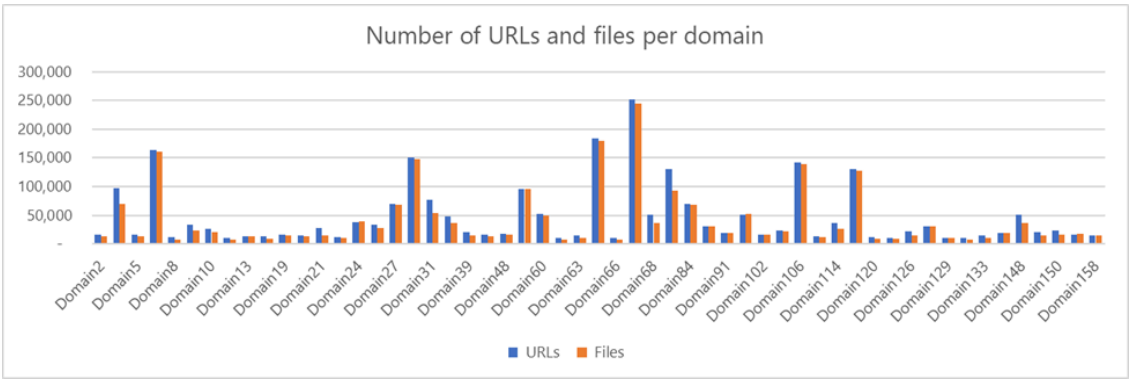


*Figure 3. Number of URLs and number of files hosted on Adrozek domains with at least 100 files.*

The distribution infrastructure is also very dynamic. Some of the domains were up for just one day, while others were active for longer, up to 120 days. Interestingly, we saw some of the domains distributing clean files like Process Explorer, likely an attempt by the attackers to improve the reputation of their domains and URLs, and evade network-based protections.

## Installation

Attackers use this sprawling infrastructure to distribute hundreds of thousands of unique Adrozek installer samples. Each of these files is heavily obfuscated and uses a unique file name that follows this format: *setup_<application name>_<numbers>.exe*.

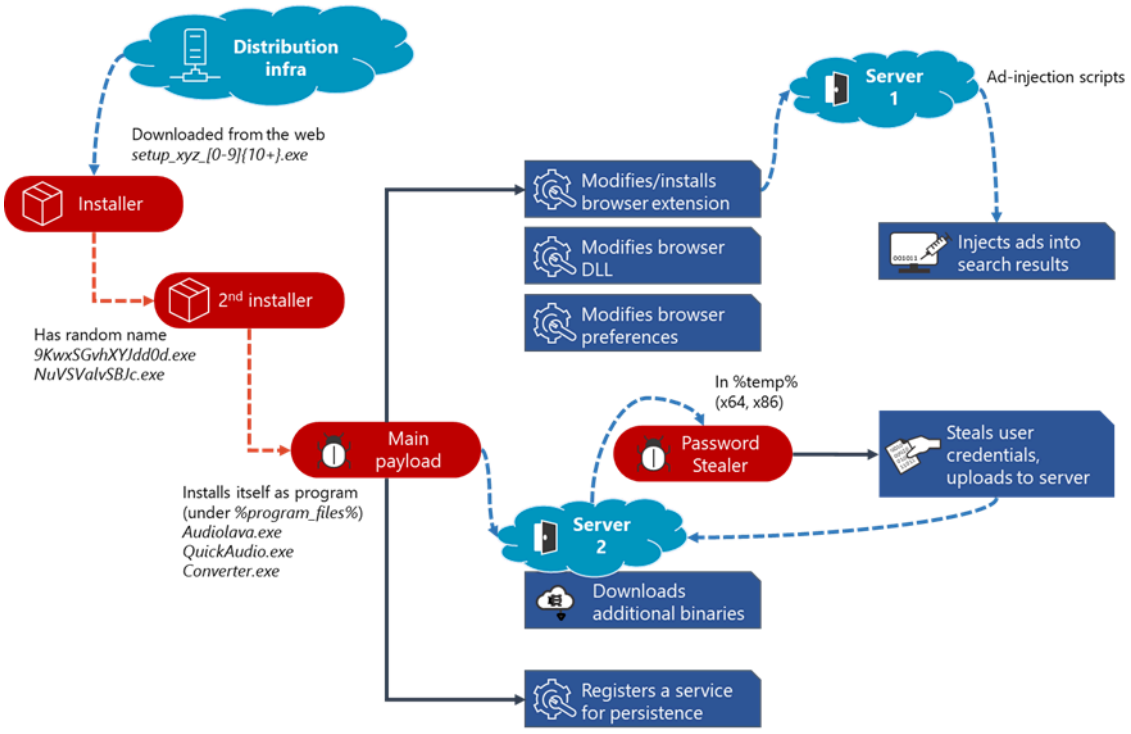


*Figure 4. Adrozek attack chain*

When run, the installer drops an .exe file with a random file name in the %temp% folder. This file in drops the main payload in the Program Files folder using a file name that makes it look like a legitimate audio-related

software. We have observed the malware use various names like *Audiolava.exe*, *QuickAudio.exe*, and *converter.exe*. The malware is installed like a usual program that can be accessed through *Settings>Apps & features*, and registered as a service with the same name.

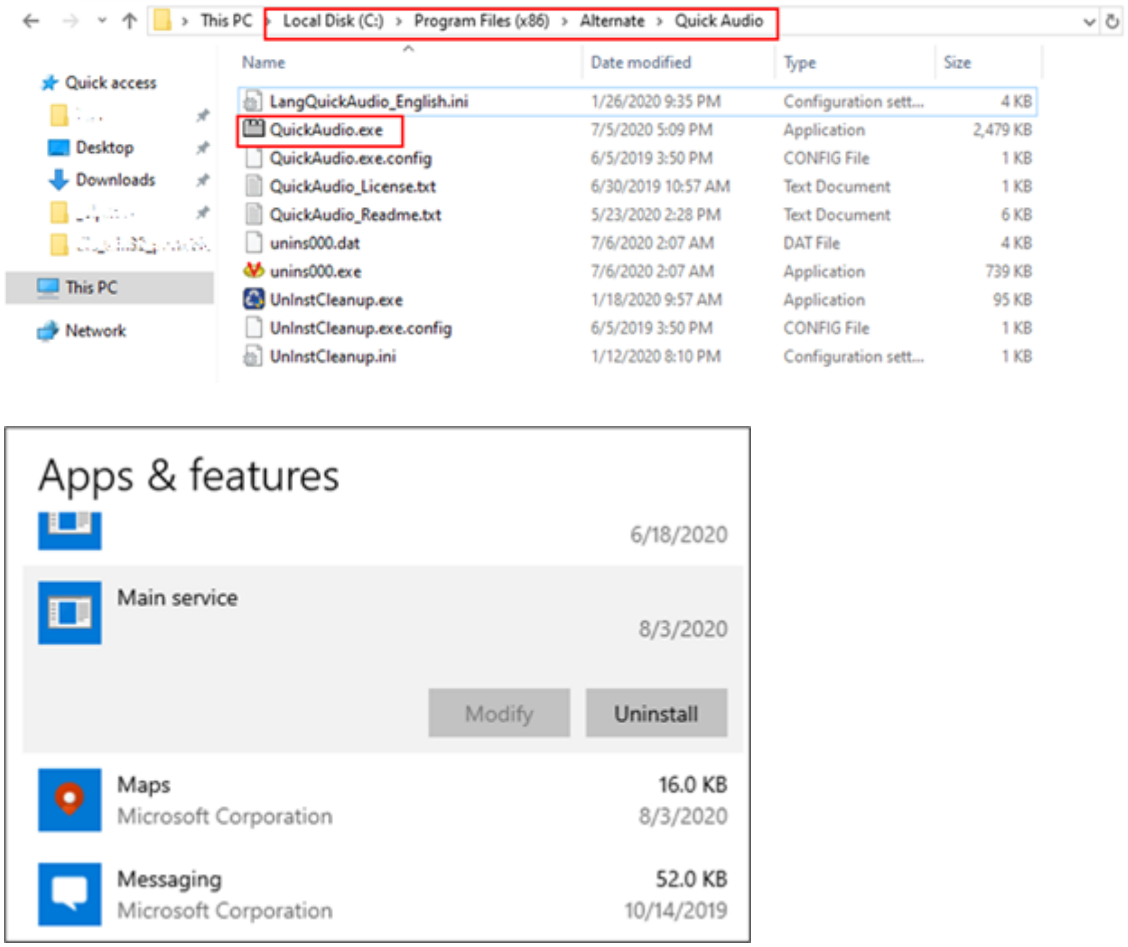


*Figure 5. Adrozek installed as a program that can be accessed through the Apps & features setting*

## Modifying browser components

Once installed, Adrozek makes multiple changes to the browser settings and components. These changes allow the malware to inject ads into search engine result pages.

### Extensions

The malware makes changes to certain browser extensions. On Google Chrome, the malware typically modifies "Chrome Media Router", one of the browser's default extensions, but we have seen it use different extensions.

Each extension on Chromium-based browsers has a unique 32-character ID that users can use to locate the extension on machines or on the Chrome Web store. On Microsoft Edge and Yandex Browser, it uses IDs of legitimate extensions, such as "Radioplayer" to masquerade as legitimate. As it is rare for most of these extensions to be already installed on devices, it creates a new folder with this extension ID and stores malicious components in this folder. On Firefox, it appends a folder with a Globally Unique Identifier (GUID) to the browser extension. In summary, the paths and extension IDs used by the malware for each browser are below:

| Browser | Extension paths examples |
|---|---|
| Microsoft Edge | *%localappdata%\Microsoft\Edge\User Data\Default\Extensions\fcppdfelojakeahklfgkjegnpbgndoch* |
| Google Chrome | *%localappdata%\Google\Chrome\User Data\Default\Extensions\pkedcjkdefgpdelpbcmbmeomcjbeemfm (might vary)* |
| Mozilla Firefox | *%appdata%\Roaming\Mozilla\Firefox\Profiles\<profile>\Extensions\ {14553439-2741-4e9d-b474-784f336f58c9}* |
| Yandex Browser | *%localappdata%\Yandex\YandexBrowser\User Data\Default\Extensions\fcppdfelojakeahklfgkjegnpbgndoch* |

Despite targeting different extensions on each browser, the malware adds the same malicious scripts to these extensions. In some cases, the malware modifies the default extension by adding seven JavaScript files and one *manifest.json* file to the target extension's file path. In other cases, it creates a new folder with the same malicious components.

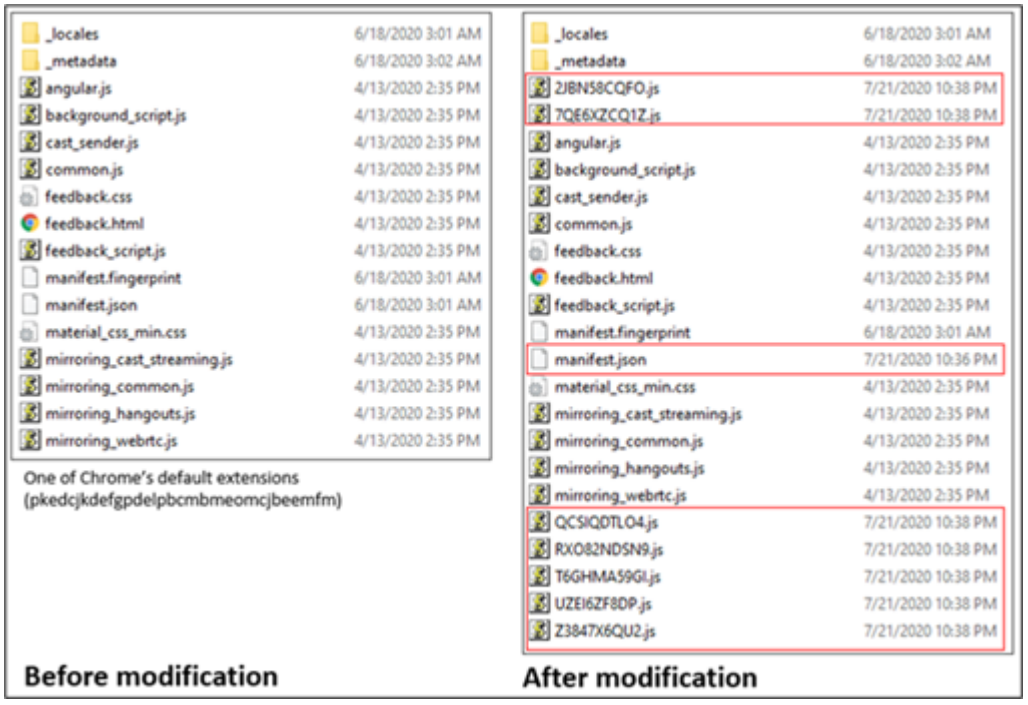


*Figure 6. JavaScript and JSON files added to the target extension's file path*

These malicious scripts connect to the attacker's server to fetch additional scripts, which are responsible for injecting advertisements into search results. The domain name of the remote server is specified in the extension's scripts. The malware also sends information about the device to the said remote server.


*Figure 7. Additional downloaded script*

## Browser DLLs

The malware also tampers with certain browser DLLs. For instance, on Microsoft Edge, it modifies *MsEdge.dll* to turn off security controls that are crucial for detecting any changes in the *Secure Preferences* file.

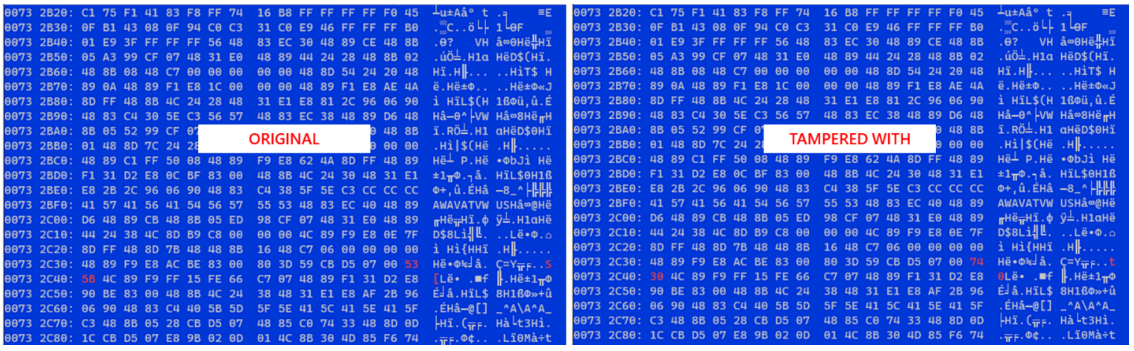


*Figure 8. Comparison of original and tampered with MsEdge.dll.*

This technique impacts not only Microsoft Edge but other Chromium-based browsers. These browsers store user settings and preferences, such as home page and default search engine, in the *Preferences* file. For each of the four target browsers, it modifies the relevant DLL:

| Browser | Modified files |
|---|---|
| Microsoft Edge | *%PROGRAMFILES%\Microsoft\Edge\Application\<version>\msedge.dll*<br>*%localappdata%\Microsoft\Edge\User Data\Default\Secure Preferences*<br>*%localappdata%\Microsoft\Edge\User Data\Default\Preferences* |
| Google Chrome | *%PROGRAMFILES%\Google\Chrome\Application\<version>\chrome.dll*<br>*%localappdata%\Google\Chrome\User Data\Default\Secure Preferences*<br>*%localappdata%\Google\Chrome\User Data\Default\Preferences* |
| Yandex Browser | *%PROGRAMFILES%\Yandex\YandexBrowser\<version>\browser.dll*<br>*%localappdata%\Yandex\YandexBrowser\User Data\Default\Secure* |

| | |
|---|---|
| | *Preferences*<br>*%localappdata%\Yandex\YandexBrowser\User Data\Default\Preferences* |
| Firefox | *%PROGRAMFILES%\Mozilla Firefox\omni.ja*<br>*%appdata%\Mozilla\Firefox\Profiles\<profile>\extensions.json*<br>*%appdata%\Mozilla\Firefox\Profiles\<profile>\prefs.js* |

### Browser security settings

Browsers have security settings that defend against malware tampering. The *Preferences* file, for example, contains sensitive data and security settings. Chromium-based browsers detects any unauthorized modifications to these settings through signatures and validation on several preferences. These preferences, as well as configuration parameters, are stored in JSON file name *Secure Preferences*.

The *Secure Preferences* file is similar in structure to the *Preferences* file except that the former adds hash-based message authentication code (HMAC) for every entry in the file. This file also contains a key named *super_mac* that verifies the integrity of all HMACs. When the browser starts, it validates the HMAC values and the *super_mac* key by calculating and comparing with the HMAC SHA-256 of some of the *JSON* nodes. If it finds values that don't match, the browser resets the relevant preference to its default value.

In the past, browser modifiers calculated the hashes like browsers do and update the *Secure Preferences* accordingly. Adrozek goes one step further and patches the function that launches the integrity check. The two-byte patch nullifies the integrity check, which makes the browser potentially more vulnerable to hijacking or tampering.

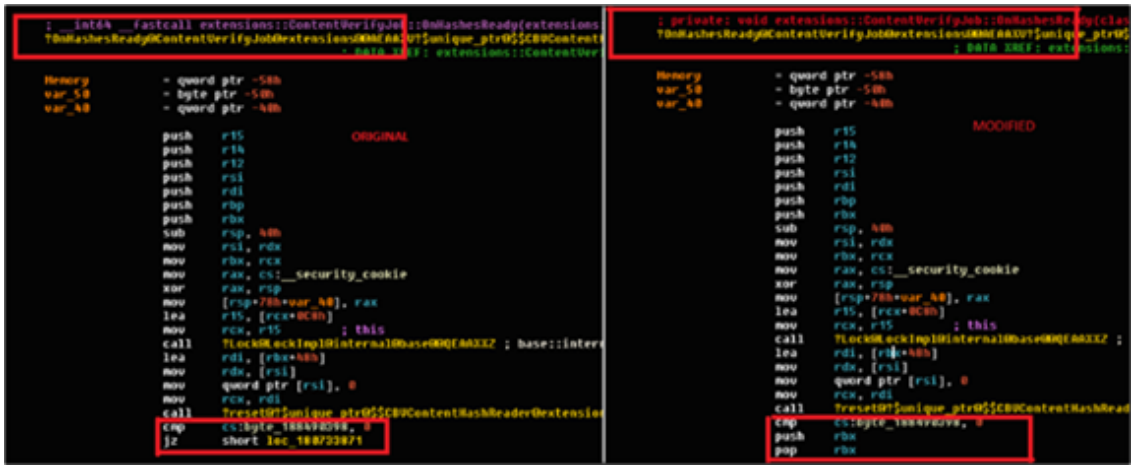


*Figure 9. Two-byte patch to the function in Secure Preferences file that launches the integrity check*

With the integrity check disabled, Adrozek proceeds to modify security settings. On Google Chrome or Microsoft Edge, the malware modifies the following entries in the *Secure Preferences* file to add permissions that enable the malicious extensions to have more control over Chrome APIs:

| **Entry in *Secure Preferences* file** | **Value** | **Result** |
|---|---|---|

| | | |
|---|---|---|
| browser_action_visible | false | Plugin not visible in the browser toolbar |
| extension_can_script_all_urls | true | Allows the extension to script on all URLs without explicit permission |
| incognito | true | The extension can run in the incognito mode |
| safebrowsing | false | Turns off safe browsing |

The screenshot below shows the permissions added to the *Secure Preferences* file:

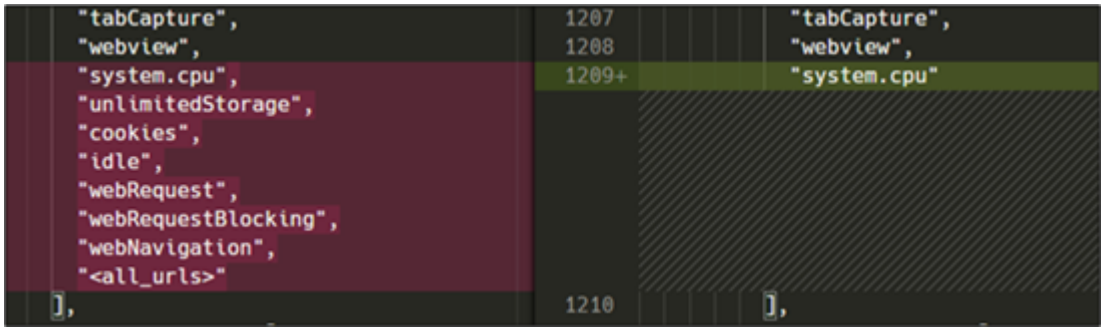


*Figure 10. Permissions added to the Secure Preferences file*

On Mozilla Firefox, Adrozek modifies the following security settings:

| **Modified file name** | **Content** | **Purpose** |
|---|---|---|
| *prefs.js* | *user_pref("app.update.auto", false); user_pref("app.update.enabled", false); user_pref("app.update.service.enabled", false)* | Turn off updates |
| *extensions.json* | *(appends details about the malicious extension)* | Register the extension to the browser |
| *Omni.ja (XPIDatabase.jsm module)* | *isNewInstall = false* | Load the extension |

## Browser updates

To prevent the browsers from being updated with the latest versions, which could restore modified settings and components, Adrozek adds a policy to turn off updates.

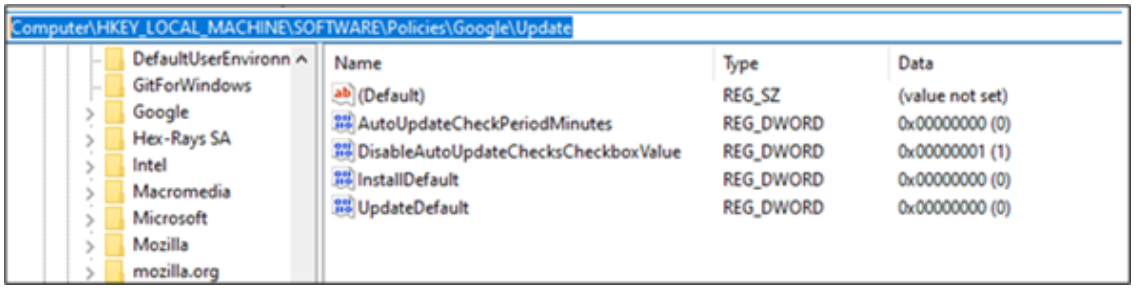


*Figure 11. Policy added to turn off updates*

## Persistence

In addition to modifying browser setting and components, Adrozek also changes several systems settings to have even more control of the compromised device. It stores its configuration parameters at the registry key *HKEY_LOCAL_MACHINE\SOFTWARE\Wow6432Node\<programName>*. The '*tag*' and '*did*' entries contain the command-line arguments that it uses to launch the main payload. More recent variants of Adrozek use random characters instead of '*tag*' or '*did*'.

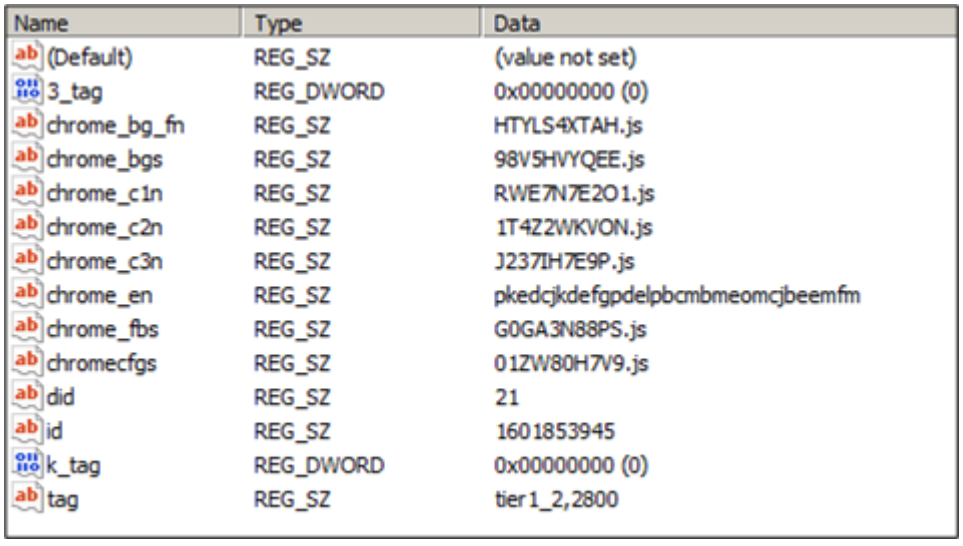


*Figure 12. Registry entries with command-line arguments that launch the main payload*

To maintain persistence, the malware creates a service named "Main Service".

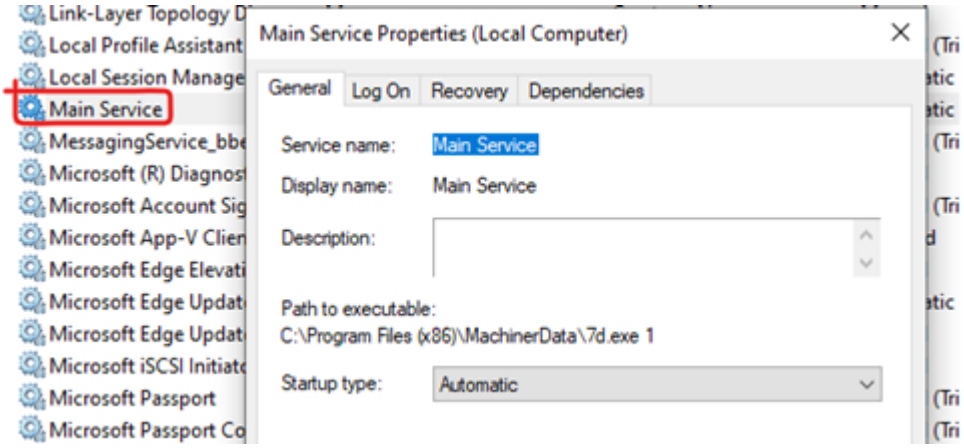

*Figure 13. Service created to maintain persistence*

## Ad injection

After tampering with multiple browser components and settings, the malware gains the capability to inject ads on search results on affected browsers. The injection of ads is performed by malicious scripts downloaded from remote servers.

Depending on the search keyword, scripts add related ads at the top of legitimate ads and search results. The number of ads inserted and the sites they point to vary. And while we have not seen these ads point to malware-hosting and other malicious sites, the attackers can presumably make that change anytime. The Adrozek attackers, however, operate the way other browser modifiers do, which is to earn through affiliate ad programs, which pay for referral traffic to certain websites.

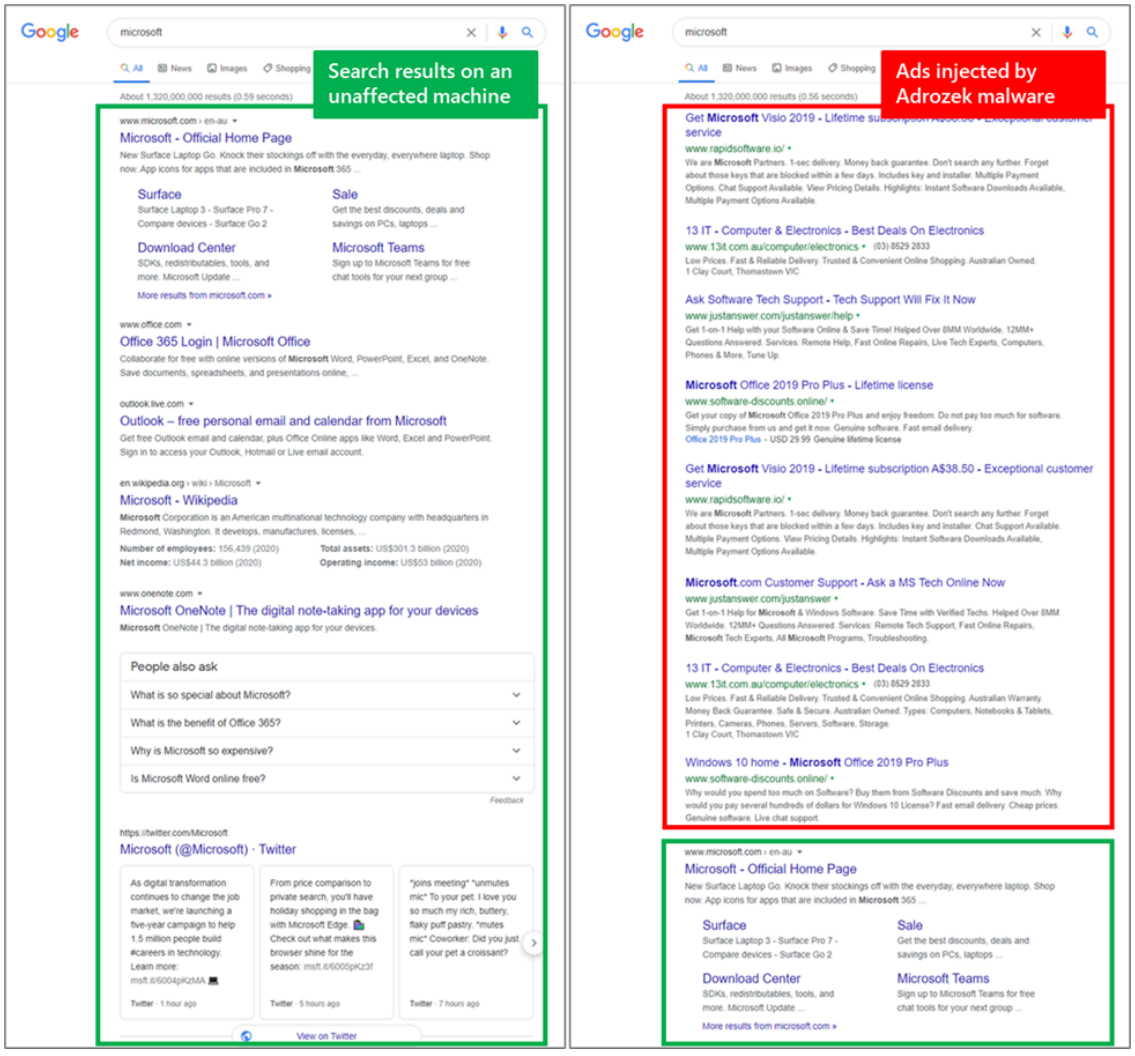


*Figure 14. Comparison of search results pages on an affected machine and one with Adrozek running*

## Credential theft

On Mozilla Firefox, Adrozek takes things further. It makes the most of its foothold by performing credential theft. It downloads an additional randomly named *.exe* file, which collects device information and the currently active username. It sends this information to the attacker.


*Figure 15. Additional executable file written to the %temp% folder*

It then starts locating specific files, including *login.json*. On Mozilla Firefox, the said file, which is located at *%appdata%\Roaming\Mozilla\Firefox \Profiles\<profile>\logins.json*, stores user credentials in encrypted form and the browsing history.

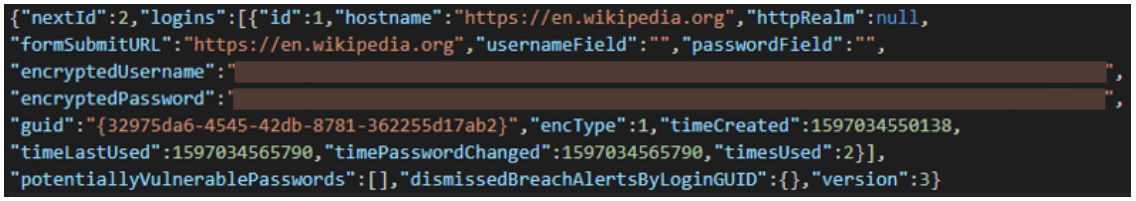


*Figure 16. JSON file containing stolen credentials*

The malware looks for specific keywords like *encryptedUsername* and *encryptedPassword* to locate encrypted data. It then decrypts the data using the function *PK11SDR_Decrypt()* within the Firefox library and sends it to attackers.

With this additional function, Adrozek sets itself apart from other browser modifiers and demonstrates that there's no such thing as low-priority or non-urgent threats. Preventing the full range of threat from gaining access in the first place is of utmost importance.

## Defending against sophisticated browser modifiers

Adrozek shows that even threats that are not thought of as urgent or critical are increasingly becoming more complex. And while the malware's main goal is to inject ads and refer traffic to certain websites, the attack chain involves sophisticated behavior that allow attackers to gain a strong foothold on a device. The addition of credential theft behavior shows that attackers can expand their objectives to take advantage of the access they're able to gain.

These complex behaviors, and the fact that the campaign uses polymorphic malware, require protections that focus on identifying and detecting malicious behavior. Microsoft Defender Antivirus, the built-in endpoint

protection solution on Windows 10, uses behavior-based, machine learning-powered detections to block Adrozek.

End users who find this threat on their devices are advised to re-install their browsers. Considering the massive infrastructure that was used to distribute this threat on the web, users should also educate themselves about preventing malware infections and the risks of downloading and installing software from untrusted sources and clicking ads or links on suspicious websites. Users should also take advantage of URL filtering solutions, such as Microsoft Defender SmartScreen on Microsoft Edge. Configuring security software to automatically download and install updates, as well as running the latest versions of the operating system and applications and deploying the latest security updates help harden endpoints from threats.

For enterprises, defenders should look to reduce the attack surface for these types of threats. Application control allows organizations to enforce the use of only authorized apps and services. Enterprise-grade browsers like Microsoft Edge provide additional security features like conditional access and Application Guard that defend against threats on the browser.

It's also important for enterprises to gain deep visibility into malicious behaviors on endpoints and the capability to correlate with threat data from other domains like cloud apps, email and data, and identities. Microsoft 365 Defender delivers coordinated protection across domains and provides rich investigation tools that empower defenders to respond to attacks. Learn how your organization can stop attacks through automated, cross-domain security and built-in AI with Microsoft Defender 365.

*Microsoft 365 Defender Research Team*

---

### Talk to us

Questions, concerns, or insights on this story? Join discussions at the Microsoft 365 Defender tech community.

Read all Microsoft security intelligence blog posts.

Follow us on Twitter **@MsftSecIntel**.

---