SocGholish | Red Canary Threat Detection Report

Archived: 2026-04-05 13:25:57 UTC

Analysis

SocGholish is a malware family that leverages drive-by-downloads masquerading as software updates for initial access. Active since at least April 2018, SocGholish has been linked to the suspected Russian cybercrime group Evil Corp. As in past years, Red Canary observed SocGholish impacting a wide variety of industry verticals in 2025.

Like previous years, SocGholish activity maintained a relatively constant background volume in 2025, with periods of higher activity followed by a slow tapering. Throughout the year, activity peaked in January/February, May, and September, with lower but steady volume during the remaining months in the year. As usual, the spikes in activity coincided with changes in lures.

Also known as “FakeUpdates,” SocGholish typically gains initial access by presenting visitors of a compromised website with a lure indicating an update is needed for their browser or other common software. Silent Push published a detailed summary of the traffic distribution systems (TDS) and web injects in early August. Red Canary visibility typically begins downstream of that activity, once a user has taken the bait. Unsuspecting users who download the “update” are tricked into running a malicious JavaScript payload, launching the attack. Historically SocGholish wrapped this JavaScript (JS) payload within a ZIP file, however, since late 2022 direct delivery of the JS without the ZIP file has also been observed.

- Analysis
- Take action
- Detection
- Testing

In 2025, about one third of SocGholish infections detected by Red Canary involved a ZIP file, while two thirds used a direct to JS lure.

Do you `C` what I `C`?

One of the distinguishing characteristics of SocGholish filename lures continues to be their use of homoglyphs. SocGholish began using these “lookalike” characters in 2022 to replace certain characters in filenames, likely in an attempt to evade detection based on filename patterns. For example, instead of the typical filename `Chrome.Update.zip`, SocGholish would replace the letters `C` and `a` with their UTF-8 Cyrillic look-alike characters `С` (`0xd0a1`) and `а` (`0xd0b0`), to produce the filename `Chrome.Update.zip`. While nearly identical in appearance to the human eye, the filenames appear different to a computer comparing strings.

SocGholish lures in 2025 picked up where they left off in 2024, using a direct JS download named `Update.js` with the homoglyph replacing the letter `p`. This lure continued through mid-January, at which point they made a subtle change by switching to a homoglyph `а` and returning to the ASCII `p` (`Update.js`). This lure continued to be used through late March. Interspersed with these direct to JS homoglyph lures, we also encountered `Update.zip` lures containing an identically named `Update.js` file with no homoglyphs present.

After a lull in activity, we observed a new lure in late April that introduced several previously unused homoglyphs–the three-byte UTF-8 characters `Ị` (UTF-8 `0xe1bb8a`, in place of a capital letter `i`) and `Ụ` (UTF-8 `0xe1bba4`, in place of a capital letter `u`), as well as the Cyrillic letter Palochka (UTF-8 `0xd380`) in place of a lowercase `L`. These characters appeared in a ZIP lure containing a JS payload, alongside homoglyph `р` in both the ZIP and JS names and homoglyph `а` in the ZIP name only of the lure `ỤpdateỊnstaIIer.zip.\Ụpdate.js`.

ASCII character	doppelgänger character	UTF-8 hex encoding	UTF-16 hex encoding
`a`	`а` (Cyrillic Small Letter A)	`d0b0`	`0430`
`C`	`С` (Cyrillic Capital Letter Es)	`d0a1`	`0421`
`e`	`е` (Cyrillic Small letter le)	`d0b5`	`0435`
`I` (capital i)	`Ị` (Latin Capitla Letter i with dot below)	`e1bb8a`	`1eca`
`l` (lower case L)	`I` (Cyrillic Letter Palochka)	`d380`	`04c0`

`o`	`ο` (Greek Small Letter Omicron)	`cebf`	`03bf`
`p`	`р` (Cyrillic Small Letter Er)	`d180`	`0440`
`U`	`Ụ` (Latin Capital Letter U with dot below)	`e1bba4`	`1ee4`

This change was short-lived, by mid-May we were predominantly seeing direct to JS lures with the name `ChromeUpdateInstaller.js`, using homoglyphs in place of the letters `o`, `p`, and both `a`'s. This lure continued for about a month until mid-June, at which point they returned to the new homoglyphs with the same ZIP name from April (`ỤpdateịnstaIIer`) coupled with various homoglyph-free names such as `Installer.js` or `Updater.js`. By mid-July, they seemed to tire of the homoglyphs again and temporarily returned to some formerly used direct to JS lures with names like `Chrome.js` or `Edge.js`.

Activity waned through August and early September, until a new version of the javascript appeared in late September. Initially this version used the homoglyph-free lure name `New Chrome available.js`, however that was quickly replaced with the classic browser-themed `firefox.js` lure name within a ZIP file named `MozillaUpdater.zip.MozillaUpdater.zip` [sic]. In early October, direct to JS lures appeared again under the name `Click to Install New Version.js`. This name was modified multiple times during October before finally settling on `New Version (CLICK).js`, which continued to be used through the end of the year.

The next step: Reconnaissance

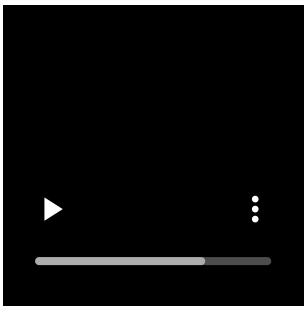
Regardless of how it is delivered, upon execution the JavaScript payload connects back to SocGholish infrastructure, where it shares details about the infected host and can retrieve additional malware. In most cases, we observe reconnaissance activity that identifies the infected endpoint and user. In rare cases, Active Directory and domain enumeration follows user discovery. The majority of SocGholish infections we detect do not progress past the reconnaissance activity, sometimes due to existing mitigations or a rapid response to isolate the host, while in other cases it appears the adversary did not progress the compromise. This likely indicates selective targeting of victims by the SocGholish adversary.

Secondary payloads

Similar to 2024, Red Canary observed a second-stage payload in about one in four SocGholish incidents in 2025. Continuing a trend from the last few years, we observed SocGholish being leveraged to deliver multiple different payloads throughout the year, likely indicative of partnering with multiple affiliate groups. While we rarely see activity beyond initial deployment of a payload, in 2025 two distinct activity clusters comprised the majority of later-stage activity following SocGholish.

Most commonly, we observed SocGholish delivering MintsLoader, which in turn deployed additional malware such as a persistent backdoor like ASyncRAT or a stealer like StealC. In addition to SocGholish, MintsLoader leveraged multiple other delivery affiliates for initial access and managed to claim its own place in the top 10 threats of 2025.

Less commonly, though perhaps of more concern, we observed SocGholish delivering a Python-based backdoor and conducting reconnaissance behaviors consistent with ransomware precursors. This activity overlaps with multiple reports from other vendors linking SocGholish to RansomHub Ransomware, a payload linked to Evil Corp among other affiliate groups. Ransomhub hasn't been observed since March 2025, according to ransomware leak site scrapers. However, the affiliates who typically deploy pre-encryption payloads are likely still operational and we assess with high confidence that, left undetected, these threats likely would have progressed to ransomware.



Take action

One of the best ways to mitigate risks associated with SocGholish, as well as Scarlet Goldfinch, Gootloader, and other threats that begin with with the malicious JavaScript execution files, is to change the default behavior in Windows to open JS files with Notepad or another editor rather than immediately executing them. Details on implementing this control via GPO are available in our blog Open with Notepad: Protecting users from malicious JavaScript.

Should SocGholish successfully execute, much of the reconnaissance conducted by the malicious JavaScript file happens in memory, with data being exfiltrated directly via POST commands to the C2 domain. One good source of insight into this behavior comes from collecting script load content, if such telemetry is available from your

endpoint detection and response (EDR) sensor. Collecting this data provides key insight into the specific commands executed and data exfiltrated.

To remove SocGholish components, stop any malicious instances of `wscript.exe`. Remove any malicious scheduled tasks for the victim user to remediate persistence on the host. If any payloads were stored within the Windows Registry or on disk, attempt to remove those payloads for full remediation.

Detection opportunities

Windows Script Host spawned from a browser and making external network connections

While JavaScript is everywhere on the web, it is rather unusual for the browser to download a JavaScript file and execute it via the Windows Script Host (`wscript.exe`). When this downloaded script starts communicating with devices outside of your network, things get even more suspicious. That said, this detection analytic may be noisy in some environments, so be prepared to identify what scripts are normally run in this way to tune out the noise.

```
 parent_process == [a browser]
&&
process == wscript.exe
&&
has_external_netconn
```

Enumerating domain trust relationships with `nltest.exe`

Left unchecked, SocGholish may lead to domain discovery. This type of behavior can be precursor to ransomware activity, and should be quickly quelled to prevent further progression of the threat.

```
 process == nltest.exe
&&
command_includes  ('/domain_trusts' || '/all_trusts')
```

- Analysis
- Take action
- Detection
- Testing

RELATED CONTENT

Source: https://redcanary.com/threat-detection-report/threats/socgholish/