red canary

SocGholish

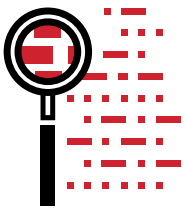
SocGholish leverages drive-by-downloads masquerading as software updates to trick visitors of compromised websites into executing malware.

PAIRS WITH THIS SONG

#5
OVERALL RANK

4.5%
CUSTOMERS AFFECTED

ANALYSIS



Analysis

Red Canary observed SocGholish impacting a wide variety of industry verticals in 2023. Similar to the spike in activity we observed in **February 2022**, in 2023 SocGholish was most active in March, suggesting a trend of increased targeting in the first quarter of the year. For the rest of the year, SocGholish maintained a relatively stable background volume, typically affecting about 0.5 percent of Red Canary-monitored environments each month.

Also known as FakeUpdates, SocGholish typically gains initial access by presenting visitors a compromised website with a lure indicating an update is needed for their browser or other common software. 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 the JS payload has been delivered directly without the ZIP cover in a majority of cases.

Do you `c` what I `c`?

Despite the shift to direct delivery of the `Update.js` file, we continued to observe a low volume of SocGholish infections that still delivered the JS within a ZIP file. In those cases, the ZIP filenames continued to follow an obfuscation trend first observed in 2022. In 2022, SocGholish began experimenting with changes to their ZIP filenames, perhaps in an attempt to evade detection based on filename patterns. During the middle of the year, SocGholish began incorporating homoglyphs (“lookalike” characters) to replace certain characters in filenames. 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`.

characters in different campaigns.

Secondary payloads

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. The majority of SocGholish infections we've detected did not result in a second-stage payload, sometimes due to existing mitigations or a rapid response to isolate the host. In most cases, we observed reconnaissance activity that only identified the infected endpoint and user. In some cases, Active Directory and domain enumeration followed user discovery. Both of these can be a precursor to lateral movement, and the cases where an additional payload was deployed often followed this additional reconnaissance. This likely indicates selective targeting of victims by the SocGholish adversary.

Consistent with the last few years, Red Canary observed a second-stage payload in about one in 10 SocGholish incidents. While historically NetSupport had been a very common payload of choice, SocGholish began showing a preference for other RATs in 2022 and this trend continued into 2023. We have not observed SocGholish delivering NetSupport since January of 2023. The first half of 2023 aligned with the latter half of 2022 wherein **Blister** with an embedded **Cobalt Strike** payload appeared most frequently. However, by the middle of the year we observed a shift to Mythic in place of Cobalt Strike, consistent with r**eporting by Fox-IT**. Within seconds of deploying an additional payload, we typically observed several post-exploitation reconnaissance behaviors often associated with **pre-ransomware activity**. SocGholish intrusions have led to various ransomware families in the past, including **Lockbit**, **WastedLocker**, and others. The adversary behind SocGholish, tracked by Proofpoint as **TA569**, is assessed to operate as an initial access broker in these cases, and may not exclusively partner with any single ransomware group.

Often imitated, never duplicated

Muddying the waters, 2023 saw a spate of new threats arise using TTPs very similar to SocGholish. **Scarlet Goldfinch** (aka **SmartApeSG**, HANEYMANEY, and ZPHP), **FakeSG** (aka **RogueRaticate**), **ClearFake**, and **FakeUpdateRU** all emerged within a few months of each other during mid-2023. Each of these threats uses JavaScript injected into compromised websites to deliver a fake update lure, much like SocGholish has done for years. And like early SocGholish, both Scarlet Goldfinch and FakeSG have shown a preference for NetSupport RAT as a payload. Despite the

footsteps, fake browser updates are certainly an initial access trend to keep an eye on. Further untangling the web of browser update threats, Proofpoint published an **article** on the state of the fake browser update landscape back in October.

TAKE ACTION

Much of the reconnaissance conducted by the malicious SocGholish 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 mitigate risks associated with the malicious JavaScript files used by SocGholish operators, we recommend preventing automatic execution of JavaScript files. You can do this by changing the default file associations for `.js` and `.jse` files. 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

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
```

Script files conducting reconnaissance with `whoami` and writing the output to a file

SocGholish employs several scripted reconnaissance commands. While much of this activity occurs in memory, one that stands out is the execution of `whoami` with the output redirected to a local temp file with the naming convention `rad<5-hex-chars>.tmp`.

```
parent_process == wscript.exe
&&
process == cmd.exe
&&
```

Enumerating domain trust relationships with `nltest.exe`

Left unchecked, SocGholish may lead to domain discovery. This type of behavior is often a 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')
```

Testing

Start testing your defenses against SocGholish using **Atomic Red Team**—an open source testing framework of small, highly portable detection tests mapped to MITRE ATT&CK.

The following tests should be sufficient to generate a useful signal for defenders in most environments:

1. **T1059.007 #2: JScript execution to gather local computer information via wscript**
2. **T1033 #6: System Discovery – SocGholish whoami**
3. **T1482 #2: Discover domain trusts with nltest**

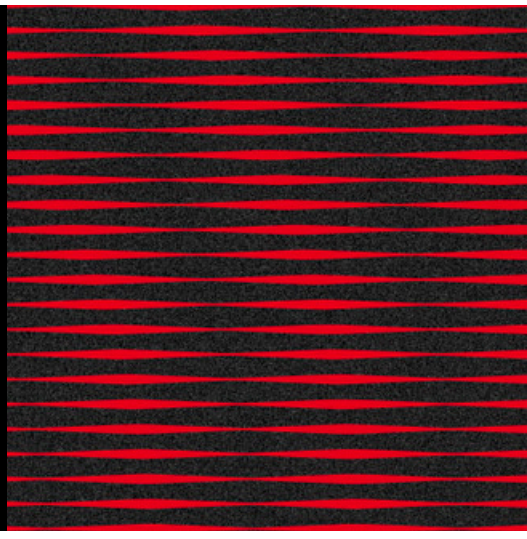
However, we wrote an entire blog about **emulating SocGholish with Atomic Red Team**, which we encourage you to read if you're interested in validating detection coverage for this threat.

Review and repeat

Now that you have executed one or several common tests and checked for the expected results, it's useful to answer some immediate questions:

- Were any of your actions detected?
- Were any of your actions blocked or prevented?
- Were your actions visible in logs or other defensive telemetry?

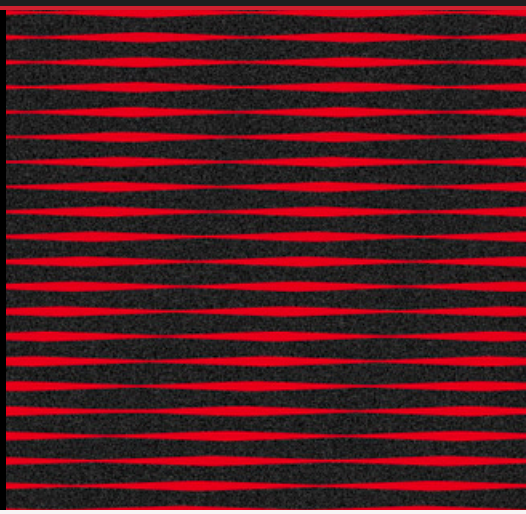
Repeat this process, performing additional tests related to this technique. You can also **create and contribute** tests of your own.



red canary

action

Schedule your demo now



Search

PRODUCTS

- Managed Detection and Response (MDR)
- Readiness Exercises
- Linux EDR
- Atomic Red Team™
- Mac Monitor
- What's New?
- Plans

SOLUTIONS

- Deliver Enterprise Security Across Your IT Environment
- Get a 24×7 SOC Instantly
- Protect Your Corporate Endpoints and Network
- Protect Your Users' Email, Identities, and SaaS Apps
- Protect Your Cloud
- Protect Critical Production Linux and Kubernetes
- Stop Business Email Compromise
- Replace Your MSSP or MDR
- Run More Effective Tabletops

Stack

Minimize Downtime with After-Hours Support

RESOURCES

View all Resources
Blog
Integrations
Guides & Overviews
Cybersecurity 101
Case Studies
Videos
Webinars
Events
Customer Help Center
Newsletter

PARTNERS

Overview
Incident Response
Insurance & Risk
Managed Service Providers
Solution Providers
Technology Partners
Apply to Become a Partner

COMPANY

About Us
The Red Canary Difference
News & Press
Careers – We're Hiring!
Contact Us
Trust Center and Security

 info@redcanary.com
+1 855-977-0686 Privacy Policy Trust Center and Security Cookie Settings