COLDRIVER Using New Malware To Steal Documents From Western Targets and NGOs

cloud.google.com/blog/topics/threat-intelligence/coldriver-steal-documents-western-targets-ngos

Google Threat Intelligence

Visibility and context on the threats that matter most.

Learn more

Written by: Wesley Shields

Google Threat Intelligence Group (GTIG) has identified a new piece of malware called LOSTKEYS, attributed to the Russian government-backed threat group COLDRIVER (also known as UNC4057, Star Blizzard, and Callisto). LOSTKEYS is capable of stealing files from a hard-coded list of extensions and directories, along with sending system information and running processes to the attacker. Observed in January, March, and April 2025, LOSTKEYS marks a new development in the toolset of COLDRIVER, a group primarily known for credential phishing against high-profile targets like NATO governments, non-governmental organizations (NGOs), and former intelligence and diplomatic officers. GTIG has been tracking COLDRIVER for many years, including their SPICA malware in 2024.

COLDRIVER typically targets high-profile individuals at their personal email addresses or at NGO addresses. They are known for stealing credentials and after gaining access to a target's account they exfiltrate emails and steal contact lists from the compromised account.

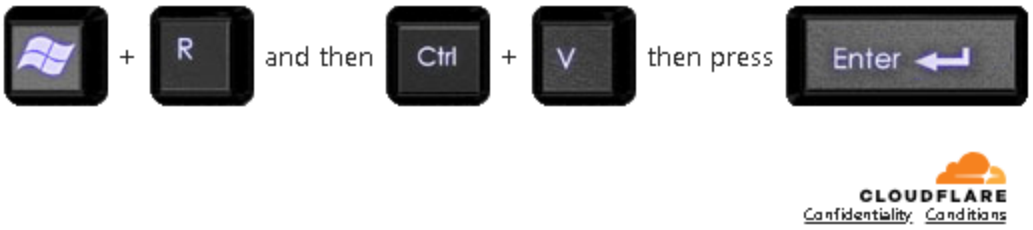
In select cases, COLDRIVER also delivers malware to target devices and may attempt to access files on the system.

Recent targets in COLDRIVER's campaigns have included current and former advisors to Western governments and militaries, as well as journalists, think tanks, and NGOs. The group has also continued targeting individuals connected to Ukraine. We believe the primary goal of COLDRIVER's operations is intelligence collection in support of Russia's strategic interests. In a small number of cases, the group has been linked to hack-and-leak campaigns targeting officials in the UK and an NGO.

To safeguard at-risk users, we use our research on serious threat actors like COLDRIVER to improve the safety and security of Google's products. We encourage potential targets to enroll in Google's Advanced Protection Program, enable Enhanced Safe Browsing for Chrome, and ensure that all devices are updated.

Stage 1 — It Starts With A Fake CAPTCHA

LOSTKEYS is delivered at the end of a multi-step infection chain that starts with a lure website with a fake CAPTCHA on it. Once the CAPTCHA has been "verified," PowerShell is copied to the users clipboard and the page prompts the user to execute the PowerShell via the "run" prompt in Windows:



The first stage PowerShell that is pasted in will fetch and execute the second stage. In multiple observed cases, the second stage was retrieved from 165.227.148[.]68.

COLDRIVER is not the only threat actor to deliver malware by socially engineering their targets to copy, paste, and then execute PowerShell commands—a technique commonly called “ClickFix.” We have observed multiple APT and financially motivated actors use this technique, which has also been widely reported publicly. Users should exercise caution when encountering a site that prompts them to exit the browser and run commands on their device, and enterprise policies should implement least privilege and disallow users from executing scripts by default.

Stage 2 — Device Evasion

The second stage calculates the MD5 hash of the display resolution of the device and if the MD5 is one of three specific values it will stop execution, otherwise it will retrieve the third stage. This step is likely done to evade execution in VMs. Each observed instance of this chain uses different, unique identifiers that must be present in the request to retrieve the next stage. In all observed instances the third stage is retrieved from the same host as the previous stages.

Stage 3 — Retrieval of the Final Payload

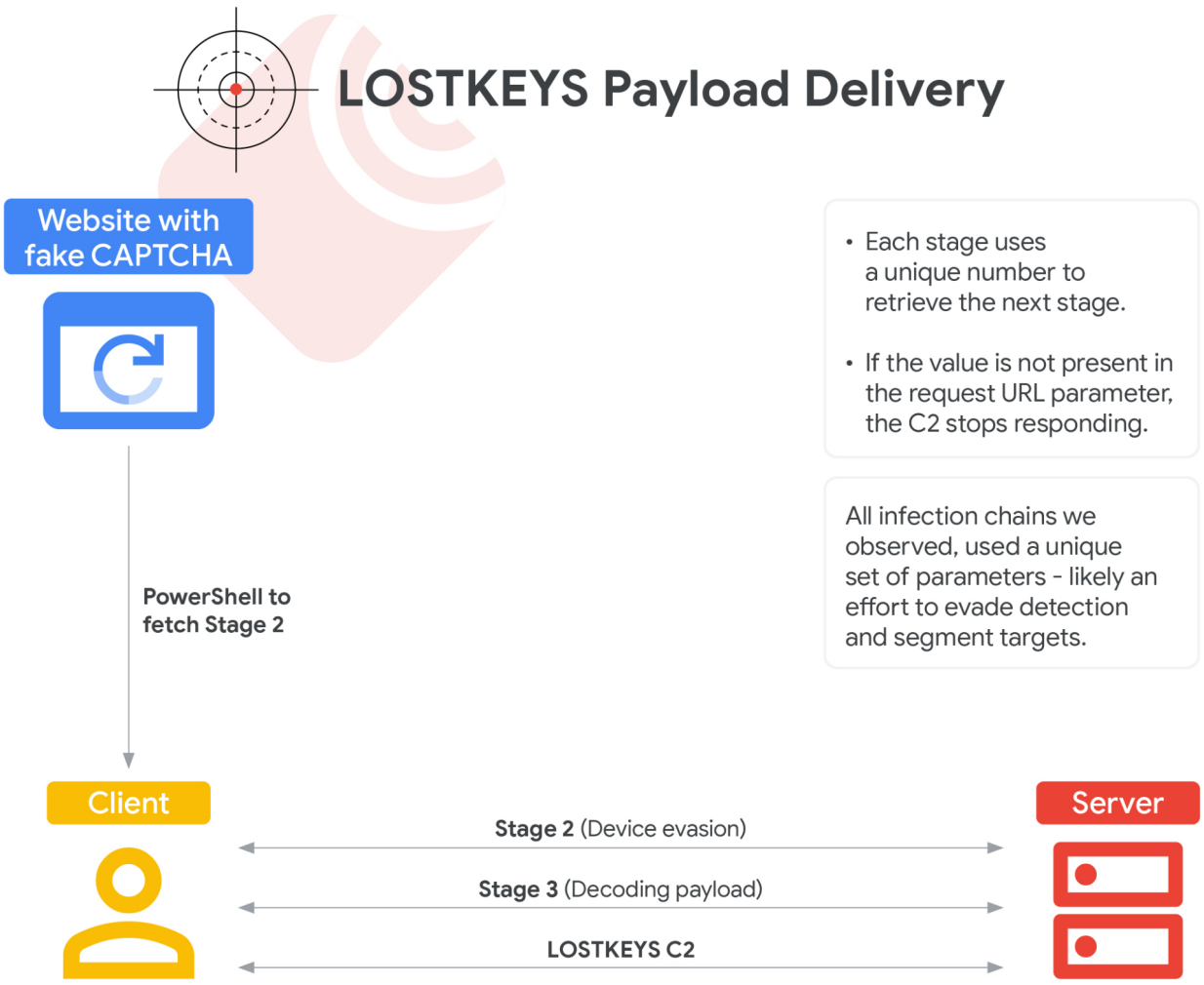
The third stage is a Base64-encoded blob, which decodes to more PowerShell. This stage retrieves and decodes the final payload. To do this it pulls down two more files, from the same host as the others, and again using different unique identifiers per infection chain.

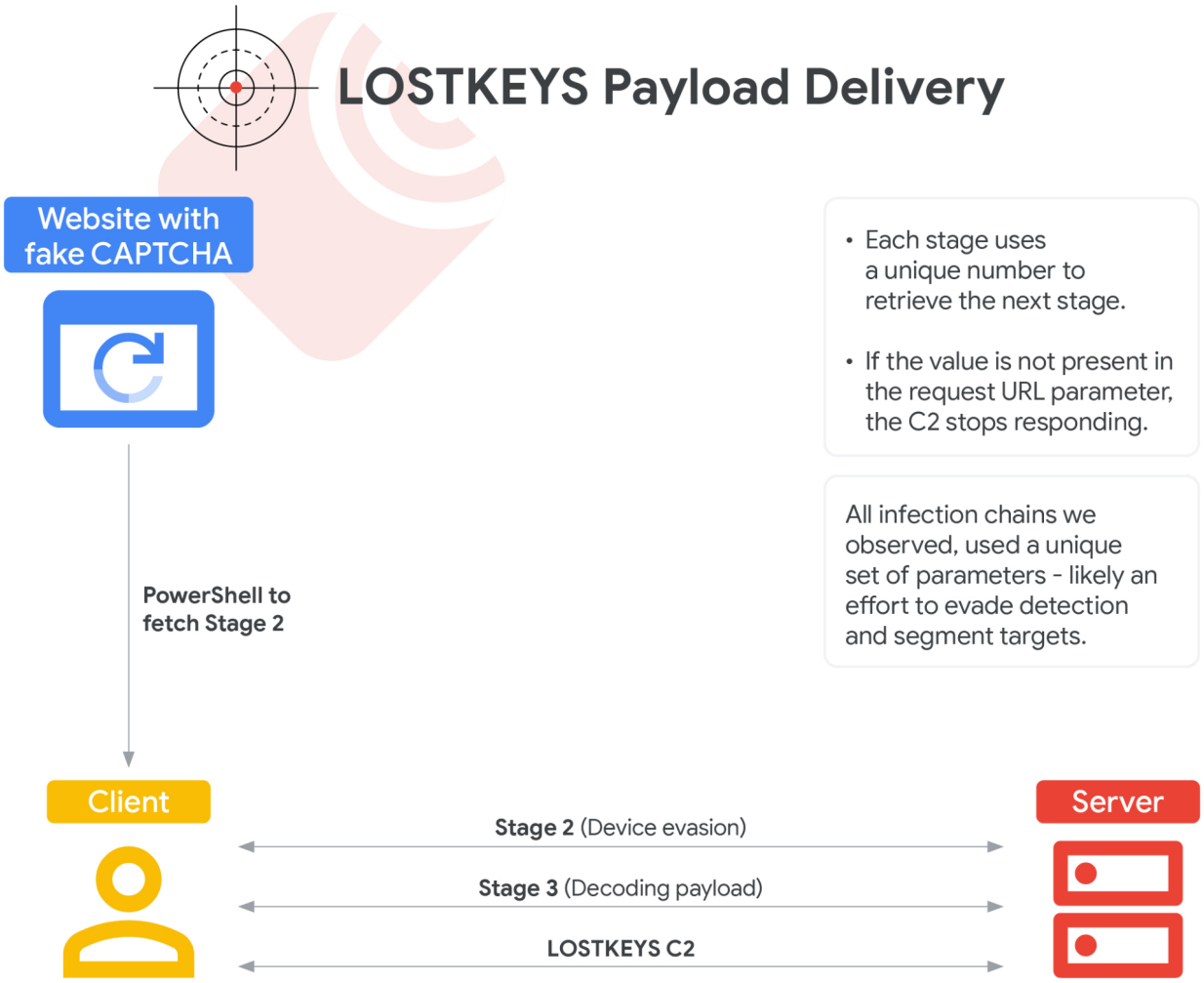
The first is a Visual Basic Script (VBS) file, which we call the “decoder” that is responsible for decoding the second one. The decoding process uses two keys, which are unique per infection chain. The decoder has one of the unique keys and the second key is stored in stage 3. The keys are used in a substitution cipher on the encoded blob, and are unique to each infection chain. A Python script to decode the final payload is:

```
# Args: encoded_file Ah90pE3b 4z7Klx1V

import base64
import sys

if len(sys.argv) != 4:
    print("Usage: decode.py file key1 key2")
    sys.exit(1)

if len(sys.argv[2]) != len(sys.argv[3]):
    print("Keys must be the same length")
    sys.exit(1)

with open(sys.argv[1], 'r') as f:
    data = f.read()

x = sys.argv[2]
y = sys.argv[3]
for i in range(len(x)):
    data = data.replace(x[i], '!').replace(y[i], x[i]).replace('!', y[i])

with open(sys.argv[1] + '.out', 'wb') as f:
    f.write(base64.b64decode(data))
```

The Final Payload (LOSTKEYS)

The end result of this is a VBS that we call LOSTKEYS. It is a piece of malware that is capable of stealing files from a hard-coded list of extensions and directories, along with sending system information and running processes to the attacker. The typical behavior of COLDRIVER is to steal credentials and then use them to steal emails and contacts from the target, but as we have previously documented they will also deploy malware called SPICA to select targets if they want to access documents on the target system. LOSTKEYS is designed to achieve a similar goal and is only deployed in highly selective cases.


LOSTKEYS Payload Delivery
Website with fake CAPTCHA
PowerShell to fetch Stage 2
Client
Server
Stage 2 (Device evasion)
Stage 3 (Decoding payload)
LOSTKEYS C2
• Each stage uses a unique number to retrieve the next stage.
• If the value is not present in the request URL parameter, the C2 stops responding.
All infection chains we observed, used a unique set of parameters - likely an effort to evade detection and segment targets.



A Link To December 2023

As part of the investigation into this activity, we discovered two additional samples, hashes of which are available in the Indicators of Compromise section, dating back as early as December 2023. In each case, the samples end up executing LOSTKEYS but are distinctly different from the execution chain mentioned here in that they are Portable Executable (PE) files pretending to be related to the software package Maltego.

It is currently unclear if these samples from December 2023 are related to COLDRIVER, or if the malware was repurposed from a different developer or operation into the activity seen starting in January 2025.

Protecting the Community

As part of our efforts to combat threat actors, we use the results of our research to improve the safety and security of Google's products. Upon discovery, all identified malicious websites, domains and files are added to Safe Browsing to protect users from further

exploitation. We also send targeted Gmail and Workspace users government-backed attacker alerts notifying them of the activity and encouraging potential targets to enable Enhanced Safe Browsing for Chrome and ensure that all devices are updated.

We are committed to sharing our findings with the security community to raise awareness and with companies and individuals that might have been targeted by these activities. We hope that improved understanding of tactics and techniques will enhance threat hunting capabilities and lead to stronger user protections across the industry.

Indicators of compromise (IOCs) and YARA rules are included in this post, and are also available as a GTI collection and rule pack.

YARA Rules

```
rule LOSTKEYS__Strings {
  meta:
    author = "Google Threat Intelligence"
    description = "wscript that steals documents and becaons system
information out to a hardcoded address"
    hash = "28a0596b9c62b7b7aca9cac2a07b067109f27d327581a60e8cb4fab92f8f4fa9"
  strings:
    $rep0 = "my_str = replace(my_str,a1,\"!\" )"
    $rep1 = "my_str = replace(my_str,b1 ,a1 )"
    $rep2 = "my_str = replace(my_str,\"!\" ,b1 )"

    $mid0 = "a1 = Mid(ch_a,ina+1,1)"
    $mid1 = "b1 = Mid(ch_b,ina+1,1)"

    $req0 = "ReqStr = base64encode( z & \";\" &
ws.ExpandEnvironmentStrings(\"%COMPUTERNAME%\") & \";\" &
ws.ExpandEnvironmentStrings(\"%USERNAME%\") & \";\" &
fso.GetDrive(\"C:\\\").SerialNumber)"
    $req1 = "ReqStr = Chain(ReqStr,\"=+/\",\",-_\")"

    $cap0 = "CapIN \"systeminfo > \"\"\" & TmpF & \"\"\"\", 1, True"
    $cap1 = "CapIN \"ipconfig /all >>  \"\"\" & TmpF & \"\"\"\", 1, True"
    $cap2 = "CapIN \"net view >>  \"\"\" & TmpF & \"\"\"\", 1, True"
    $cap3 = "CapIN \"tasklist >>  \"\"\" & TmpF & \"\"\"\", 1, True"
  condition:
    all of ($rep*) or all of ($mid*) or all of ($req*) or all of ($cap*)
}
```

Indicators of Compromise

IOC	Notes

13f7599c94b9d4b028ce02397717a1282a46f07b9d3e2f8f2b3213fa8884b029	Stage 1 - Fake CAPTCHA page, loads PowerShell to clipboard
4c7accba35edd646584bb5a40ab78f963de45e5fc816e62022cd7ab1b01dae9c	Stage 2: Device evasion and stage 3 loader
6b85d707c23d68f9518e757cc97adb20adc8accb33d0d68faf1d8d56d7840816	Stage 3: Retrieve and decode final payload, contains key “Ah90pE3b”
3233668d2e4a80b17e6357177b53539df659e55e06ba49777d0d5171f27565dd	Decoder script, contains key “4z7Klx1V”
6bc411d562456079a8f1e38f3473c33ade73b08c7518861699e9863540b64f9a	Final payload, encoded
28a0596b9c62b7b7aca9cac2a07b067109f27d327581a60e8cb4fab92f8f4fa9	Final payload, decoded
165.227.148[.]68	C2
cloudmediaportal[.]com	C2
b55cdce773bc77ee46b503dbd9430828cc0f518b94289fbfa70b5fbb02ab1847	Binary that executes LOSTKEYS from December 2023
02ce477a07681ee1671c7164c9cc847b01c2e1cd50e709f7e861eaab89c69b6f	Binary that executes LOSTKEYS from December 2023
8af28bb7e8e2f663d4b797bf3ddbee7f0a33f637a33df9b31fbb4c1ce71b2fee	LOSTKEYS from December 2023
njala[.]dev	C2 from December 2023
80.66.88[.]67	C2 from December 2023

Posted in

Threat Intelligence