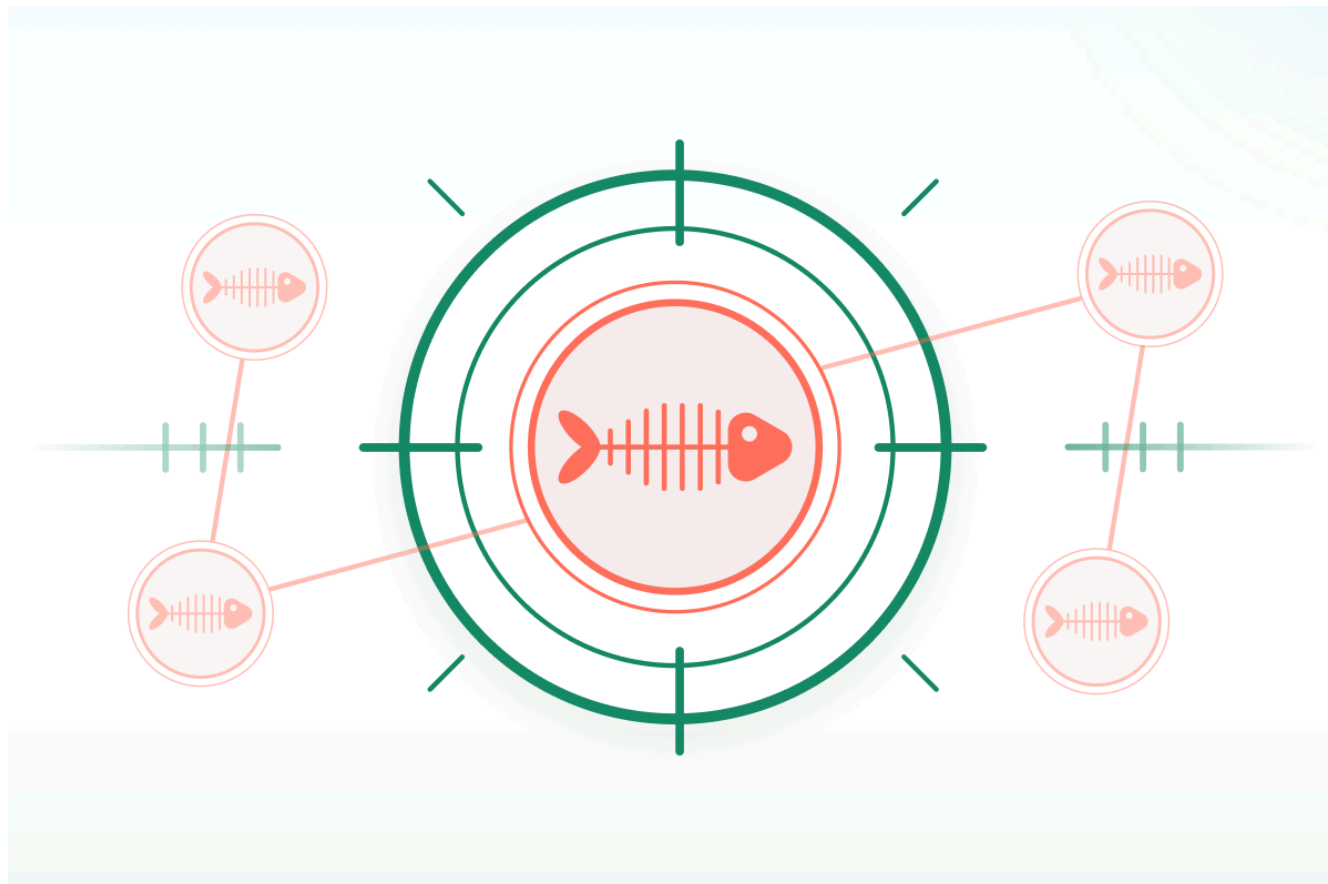


Operation Poisson Under the Spotlight

 catonetworks.com/blog/cato-ctrl-operation-poisson-analyzing-a-cybercriminals-entire-operation

Vitaly Simonovich

June 16, 2026



TL;DR

Cato CTRL recently analyzed an operator's command-and-control (C2) server's entire 33 days operation, including the steps he took to preserve access after the takedown. 339 commands. Four French victims. Between March 30 and May 1, 2026, Cato CTRL studied every command issued by a French-speaking threat actor ("Poisson") against one French automotive small business and four French individuals. With that insight, we can say with certainty, not as a prediction, that techniques like VPN-mesh-based-persistence are already in active use right now, and that taking down a C2 server is no longer sufficient for remediation.

Multi-stage fileless attack, then a keylogger. Poisson harvested banking and email credentials from real people's daily computer use using a 70-line Python keylogger he had manually retrieved. No separate exfiltration server. No beacon. Just keystrokes.

The takedown didn't end it. When we expected him to cash out, he installed OpenSSH and Tailscale VPN on a victim's machine, access that would survive even if his C2 went offline. The next day, it did; his access didn't.

Executive Summary

Cato CTRL's Vitaly Simonovich (Senior Security Researcher) has discovered a cybercriminal operation targeting French individuals and a small business. Most threat reporting is reconstructed weeks or months after the fact from malware samples and forensic leftovers. In this case, we gained detailed insight into the cybercriminal's operation. This write-up reflects a post-incident analysis based on command history and supporting telemetry captured during the March 30–May 1 window.

Poisson typed each of his 339 commands from a Havoc C2 framework server at 217[.]154[.]217[.]139[.] across 33 days, March 30 to May 1, 2026. Four victims, one French automotive small business and four French individuals. The attack utilized four Backblaze B2 payload buckets to host payloads/tools, and the mistake of an operator who accidentally left his own SSH keys and step-by-step playbook on a public storage bucket.

[2026 Cato CTRL™ Threat Report | Download the report](#)

We recovered his active SSH playbook from a public bucket and identified when he installed the persistence mechanism that would survive his C2 going offline. By analyzing the attacker's activity, Cato CTRL can state with confidence: VPN-mesh-based persistence is not theoretical. It is already being used in active intrusions. As a result, remediation can no longer stop at taking down a command-and-control server; defenders must also identify and dismantle the resilient access layer attackers leave behind.

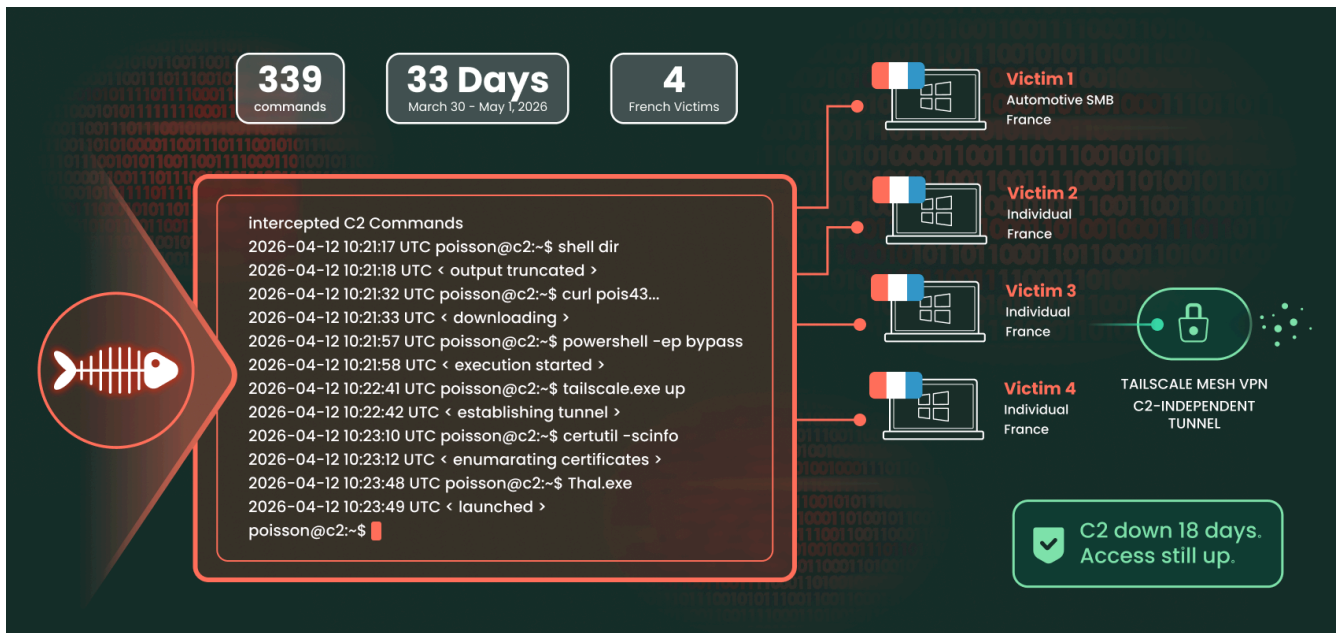


Figure 1. 339 commands over 33 days: from initial compromise to credential theft to access that survived the C2 going offline

The Operator

The operator behind the attack, handle “Poisson,” is a junior actor by any objective measure: a school-hours schedule, entirely free-tier infrastructure, OPSEC mistakes a more experienced operator would never make. And he succeeded anyway. He compromised real machines, harvested real credentials, and built persistent access that survived his C2 going offline. If this is the floor of what an untrained operator can do on free tools, the ceiling, what skilled, well-resourced actors are doing right now, is the part that should worry defenders. (Learn about how AI is lowering the bar for attackers, leading to the rise of Zero-Knowledge Threat Actor in the [2025 Cato CTRL Threat Report](#).)

His schedule fits school hours: active after 15:00 CET, with a six-hour midday gap and occasional late-night sessions. His infrastructure was free-tier across the board, DuckDNS, Backblaze B2, which hit bandwidth caps mid-operation and blocked his own downloads for hours, and a cheap IONOS VPS in Berlin. But his attack chain worked. Four machines compromised. Credentials harvested. And when his C2 went dark on April 8, his access survived through a Tailscale VPN mesh he’d installed the night before.

Technical Overview

The Operation: Day by Day

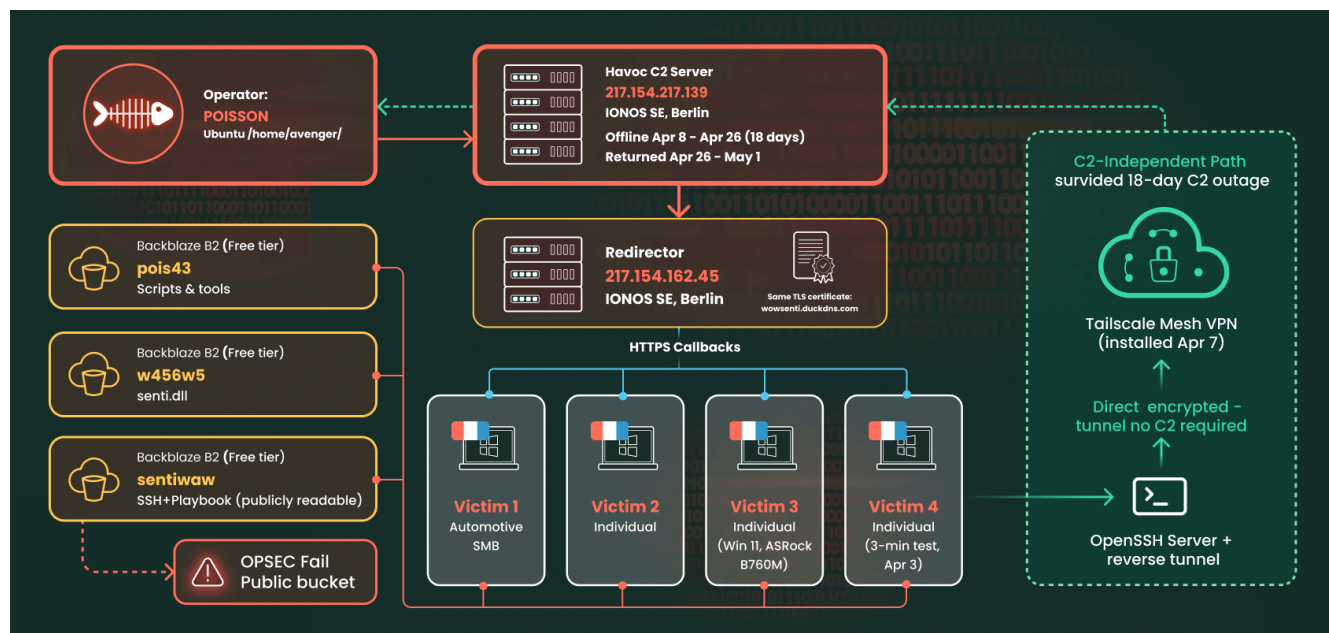


Figure 2. Poisson's infrastructure: two IONOS servers in Berlin, four B2 buckets, and a Tailscale mesh (all free-tier)

Day 1 (March 30): Victim 1 goes down in 83 minutes. The operator already had a foothold on a machine belonging to a French automotive business. He escalated to admin, deployed a scheduled task for persistence, installed a custom-compiled RustDesk remote desktop tool as a backup channel, and swapped in his own config. Done.

Day 2 (March 31): Persistence, then frustration. A quick morning session planted a startup shortcut and injected shellcode into Explorer.EXE on Victim 3, which is a French small business owner running Windows 11 on physical hardware (ASRock B760M motherboard, Intel 12th generation; confirmed not a virtual machine). That afternoon, the operator tried to escalate privileges. His approach: Start-Process -Verb RunAs. That's not a silent user account control (UAC) bypass. It pops the Windows consent dialog. Somebody has to click "Yes." Nobody did. Seven attempts. All failed.

Day 3 (April 1): 51 commands. The most active day. More failed UAC on Victim 3. He pivoted to Victim 2 to test if it worked at all (it did, as the user clicked "Yes"). Back to Victim 3. Screenshots. Registry checks. Downloaded his own script back to debug it. After a dozen tries across two days, the user finally clicked "Yes." Admin access. Scheduled task deployed. The rest of the day: consolidating access, deploying RustDesk, and adjusting power settings.

Day 4 (April 2): The keylogger changes everything. The operation shifted from “I have access” to “I’m stealing credentials.” The operator deployed KeyL.zip, a 70-line Python keylogger using pynput. It logged every keystroke to a local text file. No exfiltration server and no beacon. He retrieved it manually through Havoc and checked four times in one day. By the third check: ~3,000 characters captured. He also ran `powercfg /change standby-timeout-ac 300`, keeping the machine awake 24/7 for harvesting.

Inside the keylogger package, we found something he didn’t intend to share: a test file containing his own keystrokes. “Stikou68!!!” was typed repeatedly. The “68” matches département Haut-Rhin in Alsace. Same region as Victim 1’s business.

Days 5–6 (April 3–4): Maintenance. A single `powercfg` tweak on Day 5. On Day 6, a fourth victim briefly registered. A French name as hostname, with “95” matching département Val-d’Oise near Paris. The operator connected and killed the agent after four minutes. Testing. The operation was still growing.

Day 8 (April 7): The smartest thing he did. A 5-hour overnight session. From a third B2 bucket (“sentiwaw”; combining “senti” from his malware and “waw” from his domain), he downloaded SSH packages. Then OpenSSH Server was installed and Tailscale VPN was installed. Victim joined his Tailscale mesh. Key-based SSH auth configured. Reverse tunnel setup with `ssh -R` (SSH remote port forwarding). Both paths tested.

Now he could SSH into Victim 3 through Tailscale’s encrypted mesh. No C2 required. No public ports exposed.

Day 9 (April 8): The C2 goes dark. Brief morning session for final power settings. At 20:44 UTC, the Havoc C2 server went offline. Both teamserver and redirector were gone. But Poisson’s Tailscale and SSH runs on a completely separate network path.

The C2 went dark on April 8. It came back online on April 26 – 18 days later. See: The Return
His access survived.

The Return (April 26 – May 1)

The C2 went dark on April 8. It came back online on April 26 – 18 days later. Every victim was still there. The scheduled tasks had been firing at every boot, the SSH server had been listening, and the Tailscale mesh had been running continuously. When Poisson’s C2 came back, his agents reconnected automatically. No re-compromise needed.

Over the next five days, he ran 145 more commands. On April 30, he harvested keylogger output four times within 40 minutes. He ran `certutil -scinfo` ten times across four sessions –

systematically enumerating certificate stores and smart card information on Victim 3's machine, a pattern consistent with targeting PKI-based authentication credentials.

At 17:17 UTC on April 30, he accessed a file called Thales.zip from Prince's Downloads folder and transferred it to Victim 3's machine. He extracted it, found a .NET application called WinFormsApp1.exe, and ran it on Victim3's machine for 21 unattended minutes. He then downloaded a separate file he had pre-staged in his own payload bucket – Thal.exe, a 148 MB self-contained version of the same application bundled with the full .NET 8.0 runtime. He ran that for another 11 minutes.

After both executions completed, he spent four minutes deleting evidence: 17 files including all Thales-related artifacts, his original malware components, and data archives. He left the keylogger running. At 18:14 UTC on April 30, he issued his last recorded command. The C2 went offline on May 1 and has not returned.

What was in Thales.zip – and what those two applications did during their combined 32 minutes of execution – is the open question this operation leaves behind.

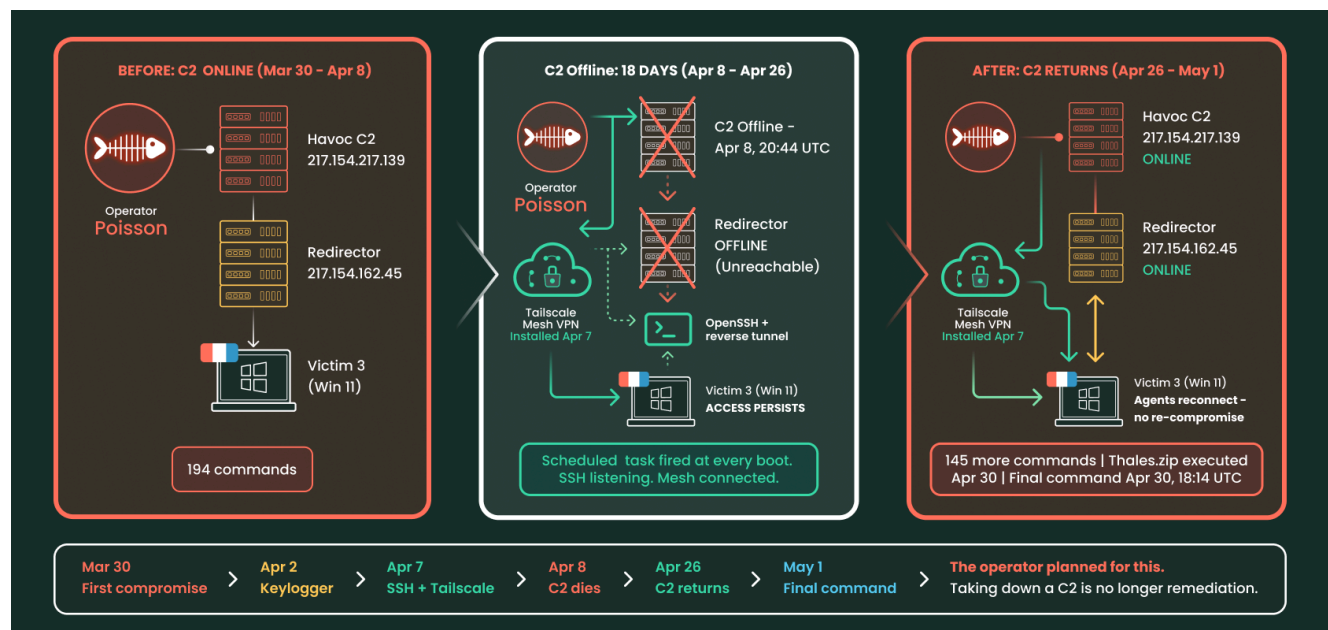


Figure 3. The C2 went offline on April 8, yet Poisson's Tailscale and SSH path kept working

The Kill Chain

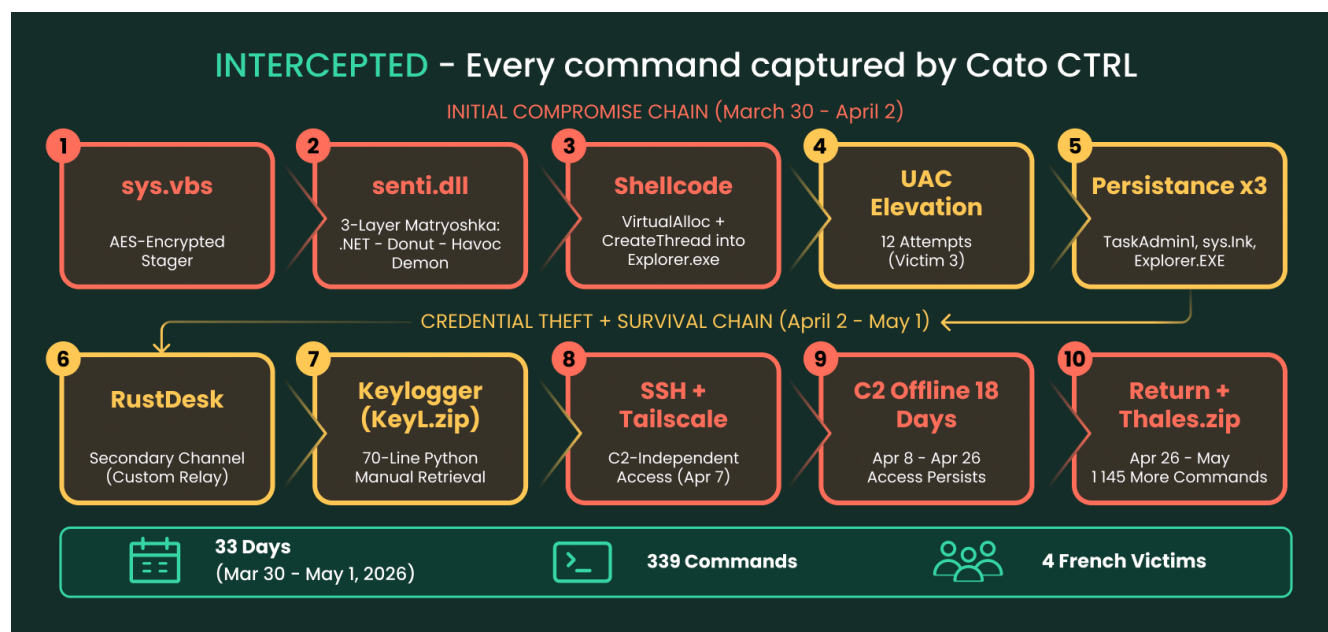


Figure 4. Seven stages: stager → fileless loader → UAC → persistence → keylogger → SSH/Tailscale

1. **sys.vbs**: AES-encrypted VBScript stager (1.1 KB). Sleeps 120 seconds (sandbox evasion) and decrypts a PowerShell payload that downloads the real malware. String split on 'Invoke'+'-Expression' to dodge static detection.
2. **senti.dll**: Four-layer matryoshka. A 3.1 MB .NET DLL encodes shellcode as 207,813 English words ("manage" → 0x4D, "therapist" → 0x48). Inside is a Donut-style reflective PE loader (XOR key=0x02) wrapping the Havoc Demon agent (XOR key=0x01). Five layers. Zero files on disk.

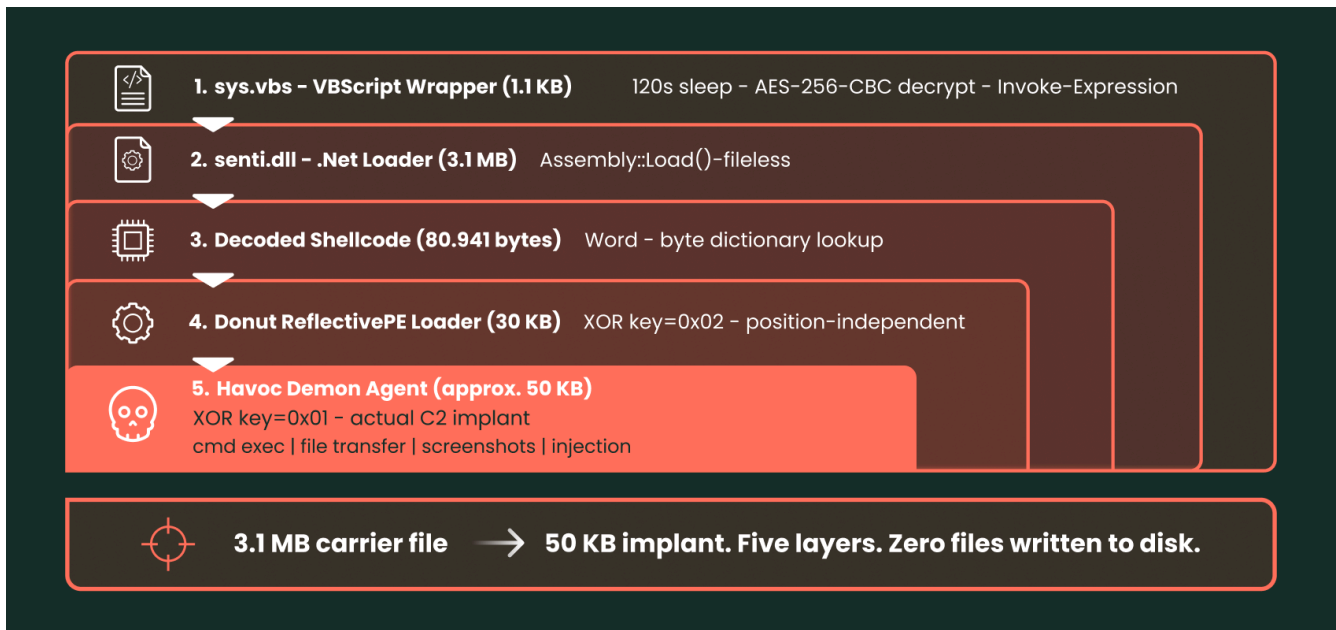


Figure 5. 3.1 MB carrier, 50 KB implant, five encoding layers, and zero files on disk

3. **UAC “bypass”:** Not a bypass but, rather, the UAC elevation attempt began with Start-Process -Verb RunAs which triggers the consent dialog. The approach worked on Victims 1 and 2 but needed a dozen attempts across two days to work on Victim 3.
4. **Persistence:** Scheduled task “TaskAdmin1” (runs stager at every logon as admin). sys.lnk in the Startup folder. Shellcode injected into Explorer.EXE.
5. **RustDesk:** Custom-compiled remote desktop with the operator’s relay config. A secondary channel independent of Havoc.
6. **Keylogger:** 70-line Python script. Local file. No beacon. Operator harvests manually. ~3,000 characters per active session.
7. **SSH and Tailscale:** OpenSSH Server and Tailscale VPN mesh. Key-based auth, reverse tunnel, and direct connection. The access that outlived the C2.

The Operator

“Poisson” isn’t an advanced persistent threat (APT). He’s almost certainly a person learning offensive security against real targets, and that is exactly the point.

His infrastructure was free-tier: DuckDNS, Backblaze B2 (which hit its bandwidth cap mid-operation), a cheap IONOS VPS in Berlin. His OPSEC was equally thin – he leaked his home directory (/home/avenger/Desktop/) five times, his bucket names trace back to his handle, and he left his complete SSH playbook, victim keys, and French installation notes on a public Backblaze bucket anyone could read. He failed at roughly half of what he tried.

He still won. Four machines compromised, real credentials harvested, C2-independent persistence that survived his own command-and-control going offline.

This is what the industry is under-pricing. Threat reporting concentrates on APTs and ransomware crews, but the long tail of low-skill operators is where small businesses and individuals – the ones with no SOC, no MDR, no threat intel – are actually getting hit. The single most consequential move in this campaign, installing Tailscale so his access would outlive his C2, was made by a student. Taking down a C2 server is no longer a remediation.

If this is the floor, the ceiling is what should worry defenders.

Protections

Cato customers are protected against attacks like the one described here through multiple layers in the [Cato SASE Platform](#):

- [Cato IPS](#) detects Havoc C2 beacon patterns and blocks known C2 infrastructure.
- Cato's threat intelligence feeds include all network indicators from this research.
- [Cato MDR](#) monitors C2 activities for fast remediation: suspicious scheduled tasks, non-standard remote access tools, shellcode injection, and hidden PowerShell execution.

For all defenders:

- Alert on OpenSSH Server (sshd) installation on Windows workstations. That's rarely legitimate on non-server machines.
- Watch for Tailscale installation and *tailscale.exe* up on machines that don't normally run VPNs.
- Detect SSH reverse tunnels (*ssh -R*) from endpoints to external servers.
- Monitor *wscript.exe* executing .vbs files from user staging directories.
- Flag scheduled tasks with *-RunLevel Highest* running script interpreters.
- Block DuckDNS subdomains.
- Watch for *powercfg /change standby-timeout* commands. Threat actors use these to keep machines awake.

Conclusion

Every piece of Poisson's kill chain served one objective: credential theft. Persistence kept the keylogger running. UAC bypass gave it admin privileges. *powercfg* kept victims awake. RustDesk and SSH/Tailscale ensured he could always retrieve captured keystrokes. Screenshots helped match passwords to services.

What he didn't do tells the same story. No Mimikatz and no lateral movement. He browsed the Downloads folder (including tax documents, insurance records, and telecom bills), but didn't steal any of them. No ransomware. No cryptominer. He doesn't care about files. He cares about what people type: banking credentials, email passwords, and government portals. For a small business owner in France, those keystrokes mean direct financial exposure.

The SSH and Tailscale move on Day 8 shows he's thinking long-term. He wasn't grabbing credentials and leaving. He was building infrastructure to maintain access even if discovered. It worked. When the C2 went down on April 8, his access to Victim 3 survived. Eighteen days later he was back – same victims, same machines, live C2. Remediation never started. The Tailscale node was still running. The SSH server was still listening. The scheduled task had been firing at every boot for six weeks.

If a junior operator on free-tier infrastructure can compromise multiple machines, build C2-independent access through a legitimate VPN, and harvest credentials from real people (all on a school schedule), how many similar campaigns are running right now? And how many will survive their C2 being taken down?

Indicators of Compromise (IoCs)

Network:

- *217[.]154[.]217[.]139* – Havoc C2 teamserver (IONOS SE, Berlin)
- *217[.]154[.]162[.]45* – C2 redirector (IONOS SE, Berlin)
- *wawsenti[.]duckdns[.]org* – TLS certificate CN for both servers
- *pois43[.]s3[.]jeu-central-003[.]backblazeb2[.]com* – payload staging (scripts/tools)
- *w456w5[.]s3[.]jeu-central-003[.]backblazeb2[.]com* – payload staging (senti.dll)
- *sentiwaw[.]s3[.]jeu-central-003[.]backblazeb2[.]com* – payload staging (SSH/playbook)

Files (SHA256):

- sys.vbs
aa7ea19e34567458b4ee66a7cd274181764984bf32123f756a7fdc64d5857b31
- senti.dll
3b7642b0f84e83a36334c608655c6cb7aae774839a6a3488526b853d89830a60
- RustCustom.zip
c79091ceae7cd592fc08e4854cda7c1182af762b6b126371cc604debd9c995fc7
- SSH.zip
f06e7e1a4363a01ba2a4fee2e28abdd623abf4194bda373f23ff0e151b5c2b45
- KeyL.zip
1f00fd604bb18bbe3081f9ce8d741c4029d2a2125eb8888ac4e0d955938059d6

- RevS.ps1
291cb1fd0f2709b4457447cbb87adacf5c36c1bcb0f8754524024d44174bb195
- Thal.exe
0378a5ef51b008aa2d6b76bd44a0bf061339bc3b737a188ec82029444d4d18fe

Host:

- Scheduled task *TaskAdmin1*: runs at logon, highest privileges
- Services: *sshd* (auto-start), *Tailscale*
- Firewall rules: *RustDesk* Full Access, *RustDesk* Full Access OUT, SSH

Operator:

- Handles: *Poisson*, *Stikou68*
- Linux: *avenger@ubuntu*, path */home/avenger/Desktop/*
- SSH key: *ssh-ed25519*
AAAAC3NzaC1lZDI1NTE5AAAAIUZH+MrtQKX4Cy68ldwV+1KKpOKU/xrdnucyG5eKZA
R avenger@ubuntu
- B2 buckets: *pois43*, *w456w5*, *sentiwaw*

Related Topics

- [Attacks, Threats, and Security News](#)
- [Cato CTRL](#)
- [Research and Engineering](#)