

Two Passwords, Two Compromise Stories — and the PhantomStealer v3.5.0 Config That Answers a Community Ask

 intel.breakglass.tech/post/two-passwords-agenttesla-phantomstealer-config

Breakglass Intelligence

April 24, 2026



MEDIUM

THREAT INTEL

Two Passwords, Two Compromise Stories — and the PhantomStealer v3.5.0 Config That Answers a Community Ask

Malware analysis · C2 infrastructure · APT attribution

@BreakGlassIntel

With thanks to [@JAMESWT_WT](#) for the original samples and for the explicit community ask to verify the PhantomStealer config, [@ShadowOpCode](#) for the partial disclosure of the corella SMTP/FTP username, and [@smica83](#) for the PowerShell dropper upload. Any mistakes below are ours, not the tipsters'. If you have prior reporting on this operator, the PhantomStealer v3.5.0 build, the GuLoader variant, or the compromise path for either host, please reach out and we'll update this post and credit the earlier source.

TL;DR

Two shared cPanel hosting accounts — one at [86.107.32\[.\]157](#) (Serverplan S.r.l., Italy) and one at [109.73.128\[.\]91](#) (Djemba IT&C SRL, Romania) — are being actively abused as FTP drop points for two separate info-stealer families. [ftp.omamontaggi\[.\]it](#) collects AgentTesla exfiltration; [ftp.corella\[.\]ro](#) collects PhantomStealer exfiltration. JAMESWT_WT surfaced both on April 22-23, 2026, with a Windows delivery chain that shipped both families from the same

MalwareBazaar campaign graph. We pulled the samples, cracked the PhantomStealer build's AES-256-CBC + PBKDF2 string encryption, and recovered the full config, including the FTP password that ShadowOpCode had held back pending verification.

The two operator-side FTP credentials tell very different stories. The AgentTesla side uses `olay@omamontaggi[.]it / pass@A12345@` — a 12-character pattern password with predictable digits, the kind of credential that shows up in commodity cred-stuffing lists or that a hosting customer sets once in 2015 and never rotates. The PhantomStealer side uses `backup@corella[.]ro / qLYMkme%hQ=S-18X` — a 16-character random-looking mixed-case string with symbols, the kind of credential a threat actor sets themselves after they already have shell on the server. Same operator running both campaigns; two different intrusion paths into two different small businesses.

The PhantomStealer sample is **version 3.5.0**, FTP-configured with SMTP, Telegram, Discord, and crypto-clipper modules all disabled. It ships as the final stage of a four-stage Windows dropper chain that smica83 uploaded as `update.ps1` on April 21: an AES-256-CBC-wrapped PowerShell decrypts to an XOR-obfuscated PowerShell, which decodes a .NET reflective launcher DLL we call `ALTERNATE.EXECUTE` (SHA-256 `c8a0077a21f2ba22ec5a6d956b012b794c8b5a70e5ccd05adcff786020850791`, not present in MalwareBazaar as of publication). The launcher hollows a legitimate Microsoft-signed LOLBin — `C:\Windows\Microsoft.NET\Framework\v4.0.30319\aspnet_compiler.exe` — and injects the PhantomStealer PE into its process space.

What This Report Adds to the Public Record

- The **full PhantomStealer v3.5.0 config** for the corella.ro build — host, username, password, mutex, enabled/disabled module flags — recovered from MalwareBazaar sample `c916f289ff9a05d74d72f28582ff03690d415fe64a4195b4f47195fe286c6d2d` by reflecting into the binary's own `Stub.StringsCrypt.DecryptConfig` method rather than reimplementing its AES+PBKDF2 key derivation.
- A **protocol correction** for defender blocklists: the corella.ro side is FTP exfiltration, not SMTP. PhantomStealer v3.5.0 contains SMTP, Telegram, Discord, and FTP code paths (it is a Stealerium-family fork); the shipped config has `SmtplibCheckBox`, `TelegramCheckBox`, `DiscordCheckBox`, and `ClipperCheckBox` all set to `0`, and `FtpCheckBox` set to `1`. ShadowOpCode's disclosed `backup@corella[.]ro` is the FTP login, not the SMTP login.
- A **new intermediate-stage IOC** — the `ALTERNATE.EXECUTE` .NET launcher DLL at SHA-256 `c8a0077a21f2ba22ec5a6d956b012b794c8b5a70e5ccd05adcff786020850791`, reconstructed from the stage-2 PowerShell by AES-decrypting the outer layer, XOR-decoding the inner layer with the embedded key `vkSecretKey765`, then base64-decoding a byte array. This launcher was not directly uploaded to MalwareBazaar; it is produced at runtime from smica83's `update.ps1`.

- A **LOLBin observation**: `aspnet_compiler.exe` used as the process-hollowing target for the PhantomStealer payload. Detection content that watches for unsigned DLLs reflectively loaded into `aspnet_compiler.exe`, or for `aspnet_compiler.exe` instances making outbound FTP connections, should flag this chain.
- **Campaign-unity evidence**: the same delivery ZIP wrapper (SHA-256 `d892b018684fb2472587c9c7ab3ae37ffd814ba3a63ab60e3b1a9f2098884c92`) and the same GuLoader binary (`Ustabil.exe`, SHA-256 `350c7cdc9d10c12ae1c490890975e387421616170f710ebbf9fa6d29fbf4b7dc`) are tagged on MalwareBazaar with both `omamontaggi-it` and `ftp-corella-ro`, meaning the AgentTesla and PhantomStealer deployments trace back to the same operator rather than two unrelated cred thefts.

This report is not attributing. We do not name the operator, do not speculate about their geography or customer base, and do not claim identification of the initial intrusion technique for either host. The two-compromise-path observation in §"" is an inference from credential quality and from the host's unpatched SSH version; it is not a proof.

The Two Hosts

Both compromised accounts are shared cPanel environments on small hosting providers that serve SMB customers in their own languages. Both sit on IP addresses registered to the hoster, not the customer. Both present a legitimate-looking WHOIS, a valid Let's Encrypt certificate, and a full cPanel/WHM/Webmail stack on the standard ports. The compromise is invisible at the HTTP layer because the website itself is unchanged — the abuse is on the FTP service only.

Host A — `86.107.32[.]157` (AgentTesla side)

Attribute	Value
IP	<code>86.107.32[.]157</code>
Reverse DNS	<code>tannen[.]dnshigh[.]com</code>
ASN	AS52030 (SERVERPLAN)
Hoster	Server Plan S.r.l. — Cassino (FR), Italy
Abuse contact	<code>abuse@serverplan[.]com</code>
TLS cert	Wildcard <code>*.omamontaggi[.]it</code> , Let's Encrypt R13, reissued 2026-04-12
FTP banner	<code>Pure-FTPd [privsep] [TLS]</code> , max 80 concurrent users
Website state	Public site returns 403 on all paths; port-80 direct IP is default cPanel redirect

Attribute	Value
Domain WHOIS	Registered 2015-06-22, legitimate small Italian industrial-assembly business
C2 URI (from AgentTesla config)	ftp://ftp.omamontaggi[.]it
FTP user (from AgentTesla config)	olay@omamontaggi[.]it
FTP password (from AgentTesla config)	pass@A12345@

The AgentTesla sample that yielded these credentials is MalwareBazaar [96adde04d7845d0bfecccab55b8892383304c0ad9b531aa08deaad632ecbad01](#), a 239 KB .NET assembly. Every config string is stored as plaintext UTF-16LE in the `.text` section; no runtime decryption is involved on the AgentTesla side. The same binary drops itself as `NKjxPLr.exe` and carries the user-agent `Mozilla/5.0 (Windows NT 10.0; Win64; x64; rv:99.0) Gecko/20100101 Firefox/99.0`.

Host B — [109.73.128\[.\]91](#) (PhantomStealer side)

Attribute	Value
IP	109.73.128[.]91
Reverse DNS	server1[.]djemba[.]ro
ASN	AS49674 (Djemba IT&C SRL)
Hoster	Djemba IT&C SRL — Arad, Romania
Abuse contact	office@djemba[.]ro
TLS cert	corella[.]ro , Let's Encrypt R12, reissued 2026-04-01
FTP banner	Pure-FTPd [privsep] [TLS], max 50 concurrent users
Website state	WordPress 6.9.4 behind a "Password Protected" plugin; customer-branded login page
SSH version	OpenSSH 8.0 (via Shodan InternetDB) — 13+ unpatched CVEs including CVE-2023-48795 (Terrapin), CVE-2020-15778, CVE-2021-41617
Co-hosted	terminustrans[.]ro (unrelated legitimate Romanian transport company, same shared host)

Attribute	Value
C2 URI (from PhantomStealer config)	<code>ftp.corella[.]ro</code>
FTP user (from PhantomStealer config)	<code>backup@corella[.]ro</code>
FTP password (from PhantomStealer config)	<code>qLYMkme%hQ=S-18X</code>

The Romanian host's SSH service is exposed on port 22 and running a version with well-known public exploit paths. That is not a claim that SSH is how the operator got in, but it is a plausible hypothesis consistent with the password quality on the PhantomStealer side (see next section).

Two Passwords, Two Doors

The clearest signal in this dataset is the difference between the two FTP passwords.

`pass@A12345@` has the shape of a hosting customer's password. It is twelve characters, it follows a predictable pattern (literal word `pass`, `@` separator, `A12345`, trailing `@`), and it is the kind of string a small-business owner might set once when cPanel prompts them for an FTP credential during onboarding and never touch again. It is also the kind of string that appears in commodity credential-stuffing wordlists and that survives twenty minutes of Hydra against a shared hosting panel. If this is the original credential the hosting customer set in 2015 and an operator harvested it from a prior stealer infection on the customer's own workstation, everything downstream makes sense: the operator pivoted to the customer's FTP account, confirmed write access, dropped AgentTesla builds pointing at it, and moved on.

`qLYMkme%hQ=S-18X` has the shape of a password generator's output. Sixteen characters, mixed case, symbols, no discernible pattern. It is not plausible that a Romanian small-business owner set this password for a `backup@` FTP account on their website; it is entirely plausible that an operator set it themselves after they already had administrative access to the hosting account. The natural next question is how they got that access. The Romanian host's SSH service runs an OpenSSH version with multiple unpatched high-severity CVEs going back to 2020; the hosting customer's original workstation could also have been popped and leaked cPanel credentials; the cPanel panel could have been brute-forced. We do not know which path was taken. We note only that the password shape on this side of the campaign is operator-authored rather than customer-authored, which pushes the intrusion further back in the kill chain than a cred-stuffing event.

Two intrusions, one operator running the downstream exfiltration. Each side of this campaign looks like a different defender conversation: the Italian side is a "stealer-log hygiene + cred rotation" conversation, the Romanian side is an "SSH patching + cPanel lockout" conversation.

Cracking PhantomStealer v3.5.0 — By Way of Its Own Decryptor

The PhantomStealer build shipped with this campaign (MalwareBazaar [c916f289ff9a05d74d72f28582ff03690d415fe64a4195b4f47995fe286c6d2d](#), 750 KB .NET assembly) does not expose its config as plaintext strings. Every sensitive value is stored as a base64 blob prefixed **ENCRYPTED:** and passed through a single decryption helper at runtime. The helper is declared as:

```
// Stub.StringsCrypt
public static string DecryptConfig(string input); // strips "ENCRYPTED:" prefix, base64-
decodes, calls Decrypt()
public static string Decrypt(byte[] bytesToBeDecrypted); // AES-256-CBC, PBKDF2 key
derivation (1000 iterations), class-field CryptKey + SaltBytes
```

On **Stub.Config**'s static constructor, every config field is initialized to an **ENCRYPTED:...** string, then an initialization routine walks the fields and calls **DecryptConfig** on each one, writing the plaintext back to the field. The routine is gated on per-feature boolean flags (**FtpCheckBox**, **SmtplibCheckBox**, **TelegramCheckBox**, etc.) — only the fields for enabled features actually get decrypted at runtime on the victim.

Rather than reverse the AES+PBKDF2 key derivation by hand and rebuild the decryption offline, we reflected into the binary and called its own **DecryptConfig** method on each static field's initial value. The process is roughly:

1. Run de4dot against the sample to rename obfuscated symbols and strip basic obfuscation (the sample was unknown-obfuscator to de4dot but the default cleanup pass is enough to reach the **Stub.Config** / **Stub.StringsCrypt** class names).
2. Disassemble the cleaned assembly to find the **DecryptConfig** signature and the list of **Stub.Config** static string fields.
3. Compile a small C# harness that **Assembly.Loads** the cleaned sample from bytes, triggers **Stub.Config**'s class constructor via **RuntimeHelpers.RunClassConstructor**, enumerates each static string field, and calls **Stub.StringsCrypt.DecryptConfig** on any value starting with **ENCRYPTED:**.
4. Run the harness under Mono.

This approach has two advantages over a hand-rolled decryption. First, it is robust to any obfuscator changes in future builds: as long as the **DecryptConfig** symbol survives deobfuscation (it does for v3.5.0, and the class name is ASCII), the harness works without needing to reverse-engineer the key material. Second, it does not execute the malware — only the single **DecryptConfig** method runs. No network, no persistence, no file drops, no keylogging. Safer than detonation, faster than full reverse engineering.

The recovered **Stub.Config** values for the corella build:

Field	Value
Version	v3.5.0
Mutex	6WWCTA0SPN0K7LMSCS01
FtpCheckBox	1 (enabled)
FtpHost	ftp.corella[.]ro
FtpUser	backup@corella[.]ro
FtpPass	qLYMkme%hQ=S-18X
SmtplibCheckBox	0 (disabled; no SMTP exfil in this build)
TelegramCheckBox	0
DiscordCheckBox	0
ClipperCheckBox	0
ChromiumBrowser	1
OutlookDesktopApp	1
FoxMailApp	1
Keylogger	0
Screenshot	0
Clipboard	0
FileZilla	0
Wifi	0
AntiAnalysis	0
Startup	0

The Stealerium lineage is visible in the field layout and in the disabled SMTP/Telegram/Discord paths — PhantomStealer v3.5.0 inherits Stealerium's modular collector-plus-exfil-channel architecture and simply does not enable most of it on this build. The operator chose FTP-only to a compromised legitimate host, which is both lower-latency than Telegram for bulk data and much lower profile for victim-side EDR (outbound FTP to a benign-looking Romanian business domain vs. outbound HTTPS to api.telegram.org).

The Delivery Chain — Four Stages From **update.ps1**

smica83 uploaded the Windows delivery vehicle to MalwareBazaar on April 21-22 as **update.ps1** (SHA-256 **c023166a028773efc229e5d4a052fd768d356f7674bc57de91169b9c47bcae55**, 3.4 MB). Reading through it, the chain decomposes as follows:

```
update.ps1 (3.4 MB PowerShell)
|
|   AES-256-CBC + PKCS7
|   key = wgJ/fzXmOQvFo6Edg9U0SoQr6rEdvegLcUT350SmDQ0= (base64, 32 bytes)
|   iv  = YzVJsAmkpoAPJnVvW5n1dA== (base64, 16 bytes)
|
▼
stage2.ps1 (XOR-obfuscated PowerShell)
|
|   XOR key = "vkSecretKey765" (14 bytes, repeating)
|   base64-encoded ciphertext embedded in script
|
▼
launcher .NET DLL (21,504 bytes, SHA-256 c8a0077a21f2...850791)
|
|   Assembly.Load in memory
|   Class: ALTERNATE.EXECUTE
|   Method: LAUNCH
|   Arguments: [aspnet_compiler.exe path, payload bytes]
|
|   Process-hollows:
|   C:\Windows\Microsoft.NET\Framework\v4.0.30319\aspnet_compiler.exe
|
▼
PhantomStealer v3.5.0 (750,592 bytes, SHA-256 c916f289...6d2d)
```

Stage 2's embedded byte array starts with **77, 90, 144, 0, 3, 0, 0, 0, 4, 0, 0, 0, 255, 255, 0, 0, ...** — an **MZ** header. This is the PhantomStealer PE itself, carried inline inside the XOR-obfuscated PowerShell. After extracting the byte array and saving it to disk, its SHA-256 matches MalwareBazaar's **c916f289...6d2d** byte-for-byte; the PhantomStealer that the in-memory loader runs is the same sample JAMESWT_WT uploaded. There is no network fetch for the final stage.

Two detection observations from the chain:

- The **ALTERNATE.EXECUTE.LAUNCH** launcher DLL is not in MalwareBazaar as a standalone sample. It is reconstituted at runtime inside a PowerShell process, from the decrypted stage-2 script. Endpoint tooling that only tracks known PE hashes will miss it. A memory-scanning rule keyed on the class name **ALTERNATE.EXECUTE** or the method name **LAUNCH**, or on the specific string **vkSecretKey765**, will catch the stage-2 script even if the PowerShell invocation itself is obfuscated.

- The process-hollowing target is `aspnet_compiler.exe`, a legitimate Microsoft-signed binary that ships with the .NET Framework. `aspnet_compiler.exe` initiating outbound FTP connections, or hosting unsigned reflectively-loaded DLLs, is a high-signal deviation from normal system behavior. Sysmon Event ID 8 (`CreateRemoteThread`) and Event ID 10 (`ProcessAccess`) with `aspnet_compiler.exe` as the target, together with Event ID 1 showing `powershell.exe` as the initiator in the same session, should light up on this chain.

Campaign Unity — One Operator, Two Families

Three pieces of evidence place both halves of this campaign under a single operator.

First, the shared MalwareBazaar tag graph. The campaign ZIP wrapper (`d892b018...4c92`) and the GuLoader sample (`350c7cdc...b7dc`, `Ustabil.exe`) are both tagged with both `omamontaggi-it` and `ftp-corella-ro`. MalwareBazaar tags propagate from submitter context; a tag graph that spans both halves of the campaign means the submitter observed both halves in the same operational context.

Second, the shared delivery vehicle. `smica83`'s `update.ps1` is tagged with `ftp-corella-ro`, `omamontaggi-it`, and `PhantomStealer` simultaneously. The script ends with `Invoke-Expression $scriptContent` after AES-decrypting the outer layer — meaning the same PowerShell wrapper could drop either stealer depending on which XOR-encrypted inner payload is embedded. Same delivery framework, pluggable final stage.

Third, the near-identical submission timeline. All six related samples (the two stealer finals, the AgentTesla packer, the GuLoader, the campaign ZIP, and the PowerShell dropper) were submitted to MalwareBazaar between April 21 06:27 UTC and April 23 06:45 UTC — a 48-hour window, by two researchers (`JAMESWT_WT` and `smica83`) who were clearly working the same cluster.

None of that individually proves single-operator control, and we are not making a harder attribution claim than the evidence supports. But "two unrelated cred thefts that happened to share a delivery ZIP, a GuLoader build, and a 48-hour submission window" is the less likely reading.

Hoster Context — Serverplan and Djemba

Both hosters run clean legitimate infrastructure. Neither is bulletproof; neither shows up in historical abuse.ch patterns as a persistent C2 substrate. The two Italian SPF-linked IPs `185.81.2[.]108` (reverse DNS `static-108-2-81-185-host[.]sphostserver[.]com`) and `185.81.4[.]150` (reverse DNS `s103[.]powermailhost[.]com`) are Serverplan's own managed mail infrastructure and are not implicated — they appear in the `omamontaggi` SPF record because Serverplan relays that customer's mail, not because the operator placed them there.

This is the compromise-of-legitimate-infrastructure pattern that's been increasingly common in 2026 commodity stealer operations: rent nothing, host nothing, own nothing. Find hosting accounts with weak credentials or exploitable edge services, repurpose them as exfiltration drop-boxes, and let the hosters absorb the abuse-report load. Blocklisting these two IPs is correct defender action, but it will not stop the operator — they will cycle to the next pair of compromised SMB hosting accounts in a week or two.

The more durable defender move is detecting the delivery chain (stage-2 XOR-PS, `ALTERNATE.EXECUTE.LAUNCH`, `aspnet_compiler.exe` hollowing) and the PhantomStealer v3.5.0 binary itself, rather than the per-campaign C2 infrastructure.

IOCs

Defanged; block as-is or normalize per your toolchain.

Network

86.107.32[.]157	AgentTesla FTP C2 (omamontaggi host)
109.73.128[.]91	PhantomStealer FTP C2 (corella host)
ftp.omamontaggi[.]it	AgentTesla FTP endpoint
omamontaggi[.]it	AgentTesla campaign domain
ftp.coreella[.]ro	PhantomStealer FTP endpoint
coreella[.]ro	PhantomStealer campaign domain
tannen[.]dnshigh[.]com	Serverplan cPanel hostname (AgentTesla side)
server1[.]djemba[.]ro	Djemba cPanel hostname (PhantomStealer side)
21/tcp	FTP exfiltration port (both hosts)

Credentials recovered from samples (rotate if observed anywhere else)

olay@omamontaggi[.]it	: pass@A12345@	– AgentTesla FTP login
backup@coreella[.]ro	: qLYMkme%hQ=S-18X	– PhantomStealer FTP login

File SHA-256

96adde04d7845d0bfecccab55b8892383304c0ad9b531aa08deaad632ecbad01	AgentTesla (final .NET, 239 KB)
3dac52fb06fdd36e5aa4fd572b2a05d6cf3d736ec8a2d487e44b828821a7ba3d	AgentTesla (packed outer, 1.1 MB)
c916f289ff9a05d74d72f28582ff03690d415fe64a4195b4f47195fe286c6d2d	PhantomStealer v3.5.0 (final .NET, 750 KB)
350c7cdc9d10c12ae1c490890975e387421616170f710ebbf9fa6d29fbf4b7dc	GuLoader "Ustabil.exe" (382 KB, both-tags)
d892b018684fb2472587c9c7ab3ae37ffd814ba3a63ab60e3b1a9f2098884c92	Campaign ZIP wrapper (231 KB, both-tags)
c023166a028773efc229e5d4a052fd768d356f7674bc57de91169b9c47bcae55	update.ps1 dropper (3.4 MB)
c8a0077a21f2ba22ec5a6d956b012b794c8b5a70e5ccd05adcff786020850791	
ALTERNATE.EXECUTE.LAUNCH .NET DLL (NEW – not in MB)	

Host-based

NKjxPLr.exe	AgentTesla drop filename / mutex
artifact	
6WWCTAOSPN0K7LMSCS01	PhantomStealer mutex
C:\Windows\Microsoft.NET\Framework\v4.0.30319\aspnet_compiler.exe	Process-hollowing target
ALTERNATE.EXECUTE	.NET launcher class name (for memory scanning)
LAUNCH	launcher method name
vkSecretKey765	XOR key in stage-2 PowerShell

User-agent (AgentTesla)

Mozilla/5.0 (Windows NT 10.0; Win64; x64; rv:99.0) Gecko/20100101 Firefox/99.0

Detection Guidance

Network

- Egress-block **86.107.32[.]157** and **109.73.128[.]91** on port 21; log and alert on any attempt. Legitimate FTP traffic to Italian or Romanian shared-hosting ranges is uncommon from most corporate environments.
- Alert on outbound FTP (port 21) from any endpoint that is not an FTP-client-using business process. AgentTesla and this PhantomStealer build both push stolen data as text files with filenames containing victim hostname, username, and timestamp.

Host

- Sysmon / EDR rule: `aspnet_compiler.exe` with outbound network connections, unsigned module loads, or parent-process `powershell.exe`. `aspnet_compiler.exe` is rarely used in enterprise environments outside developer machines running ASP.NET build pipelines.
- Memory-scan rule for the string `ALTERNATE.EXECUTE` or the method-export name `LAUNCH` inside loaded .NET assemblies in PowerShell processes — both strings are specific enough to be high-signal.
- Detection on PowerShell scripts containing the literal string `vkSecretKey765` or the AES key-base64 prefix `wgJ/fzXm0QvFo6Edg9U0SoQr6rEdvegLcUT` as hard-coded constants.

YARA

```
rule AgentTesla_FTP_omamontaggi_2026Q2 {
    meta:
        description = "AgentTesla sample configured with ftp.omamontaggi.it FTP C2"
        date = "2026-04-24"
        reference =
"https://bazaar.abuse.ch/sample/96adde04d7845d0bfecccab55b8892383304c0ad9b531aa08deaad632e
cbad01/"
    strings:
        $host = "ftp.omamontaggi.it" ascii wide nocase
        $user = "olay@omamontaggi.it" ascii wide nocase
        $uri = "ftp://ftp.omamontaggi.it" ascii wide nocase
    condition:
        uint16(0) == 0x5A4D and any of them
}

rule PhantomStealer_v3_5_0_coreella_launcher_2026Q2 {
    meta:
        description = "PhantomStealer v3.5.0 + ALTERNATE.EXECUTE launcher artifacts"
        date = "2026-04-24"
    strings:
        $host = "ftp.coreella.ro" ascii wide nocase
        $mutex = "6WWCTAOSPN0K7LMSCS01" ascii wide
        $cls = "Stub.StringsCrypt" ascii wide
        $enc = "ENCRYPTED:" ascii wide
        $lc1 = "ALTERNATE.EXECUTE" ascii wide
        $lc2 = "LAUNCH" ascii wide
        $xor = "vkSecretKey765" ascii wide
    condition:
        uint16(0) == 0x5A4D and (
            $host or $mutex or
            (2 of ($cls, $enc)) or
            (1 of ($lc1, $lc2) and $xor)
        )
}
```

Disclosure

Both hosting accounts are legitimate small businesses whose infrastructure is being abused by a third party; the IOC tables above are available for immediate blocklisting, and the hosters and their upstreams are the right parties to actually lock the accounts down. Points of contact for defenders who want to file:

- Server Plan S.r.l. (Italy) — [abuse@serverplan\[.\]com](mailto:abuse@serverplan[.]com)
- Djemba IT&C SRL (Romania) — [office@djemba\[.\]ro](mailto:office@djemba[.]ro)
- CERT-IT (Italian national CERT) — for the Italian infrastructure
- CERT-RO (Romanian national CERT) — for the Romanian infrastructure, noting the host's OpenSSH 8.0 exposure

Tipster credit: [@JAMESWT_WT](#) (original samples + config verification ask), [@ShadowOpCode](#) (partial cred disclosure), [@smica83](#) (PowerShell dropper upload). If you have prior reporting on this operator, the v3.5.0 PhantomStealer build, or either host's initial intrusion path, please reply or DM — we will update the post and credit the earlier source.

Share 

Related Reports

[high](#)

[Multi-RAT Operation Dismantled: WaterHydra APT Nexus, Five AES Keys Recovered, and Live C2 Infrastructure Mapped Across Three Continents](#)

[3/7/2026](#)