

Vidar v1.5 in Go: same family, new language, heavy sandbox checks

 derp.ca/research/vidar-go-sandbox-dead-drop

Kirk

May 16, 2026



Vidar is a name most infostealer trackers know well -- an Arkei descendant that has been snatching browser credentials and crypto wallets since 2018. It usually ships as a .NET binary or a C++ PE. The v1.5 sample we pulled from Triage on May 13, 2026 is neither. It is a 7 MB Go 1.25.4 native PE with a twelve-category sandbox scoring system, dead-drop C2 via Telegram and Steam profile pages, and enough crypto primitives to make a librarian blush.

Previous coverage of Go-based Vidar builds (Datadog's MUT-4831, Malwarebytes March 2026) established the Telegram/Steam C2 discovery pattern. Our sample follows the same strategy but belongs to a separate build track: unique botnet ID, separate C2 IP at a Hetzner box in Finland, and its own Telegram handle and Steam profile.

Sample overview

Field	Value
SHA256	2995ffb73342453b258926ec865c724e3567eee1bb8eb35d61796ee0c4f25105
SHA1	488d2dd8768e3b804179e7f0cdcebd0a7eec52b3

Field	Value
MD5	87332fcdf79e1c0bfb7713e9a52c0313
Type	PE32+ GUI (x86-64), Go 1.25.4
Size	7,211,168 bytes
Triage	10/10 (vidar)
Submitted	2026-05-13
Download filename	2.exe
Family rule	Vidar_v1_5
Botnet ID	702ef1b4007f07887e9faaee0667b50b
Version	1.5

Previous coverage

Vidar is not new, but Go-based Vidar is newer coverage:

- **Datadog Security Labs (MUT-4831)** -- Documented a Go-based Vidar variant delivered through trojanized npm packages (Oct 2025). Used Telegram and Steam URL discovery matching our sample's pattern, with different handles: telegram[.]me/sre22qe, steamcommunity[.]com/profiles/76561198777118079.
- **Trend Micro** -- Covered Vidar 2.0 (Oct 2025), a full C language rewrite with multithreaded architecture and Chrome AppBound encryption bypass. Dev alias "Loadbaks". That version track is a separate codebase from our 1.5 Go build.
- **Malwarebytes (March 2026)** -- ClickFix campaigns via compromised WordPress sites delivering Vidar. Identical Telegram/Steam C2 pattern: telegram[.]me/dikkh0k, telegram[.]me/pr55ii, Steam profiles 76561198742377525, 76561198735736086.
- **Intrinsec (April 2026)** -- Broad survey of Vidar's position in the infostealer market after the Lumma/Rhadamanthys takedowns.

Our sample sits on an older version track (1.5) with its own infrastructure. The Telegram/Steam dead-drop pattern is shared across multiple Vidar campaigns.

Not your father's .NET stealer

Capa identified a collection of crypto primitives that go beyond what a typical Vidar build needs for config decryption alone:

- AES-NI (hardware aesenc instructions)
- ChaCha20 / Salsa20 sigma constant
- RC4 PRGA implementation
- Base64 encoding
- MurmurHash3

The binary also walks PE export tables and enumerates PE sections at runtime -- capability profiles typical of reflective loading or injection, not passive data theft. Rbin confirmed the sample resolves native APIs dynamically via GetProcAddress against KERNEL32.DLL, NTDLL.DLL, WINHTTP.DLL, CRYPT32.DLL, and BCrypt.DLL, among others.

Twelve categories of "not a sandbox"

This sample has a serious suspicion of analysis environments. It implements a scored sandbox detection system: 12 checks across multiple categories, requiring 6 out of 9 core checks to pass or the binary self-terminates.

Check	What it tests
internet	Network connectivity
debugger	Debugger present
peb_flags	PEB BeingDebugged flag
cpus	CPU core count
rdtsc	RDTSC timing delta
modules	Loaded module inspection
ram	Physical RAM size (GB)
disk	Disk size (GB)
user	Username blacklist
av_sandbox	AV presence detection

Check	What it tests
pc	Hostname blacklist
uptime	System uptime

The username blacklist targets common sandbox accounts: John, sandbox, WDAGUtilityAccount. The hostname blacklist includes JOHN-PC and SANDBOX.

On top of the scoring system, the binary calls `NtSetInformationThread` with the `HideFromDebugger` flag -- a well-known anti-debugging technique.

AV kill list

The sample enumerates running processes and compares against a hardcoded blacklist of 30+ antivirus and security products. If any are detected, the sandbox score is affected. The process names checked:

AvastSvc.exe, aswEngSrv.exe, AvastUI.exe, avgcsrva.exe, avgsvc.exe, avgui.exe, ekrn.exe (ESET), egui.exe (ESET), essod.exe (ESET), PccNTMon.exe (Trend Micro), TMBMSRV.exe (Trend Micro), TmListen.exe (Trend Micro), NTRTScan.exe (Trend Micro), TmCCSF.exe (Trend Micro), coreServiceShell.exe (McAfee), avp.exe (Kaspersky), avpui.exe (Kaspersky), kavfs.exe (Kaspersky), MsMpEng.exe (Defender), MpCmdRun.exe (Defender), bdagent.exe (BitDefender), bdservicehost.exe (BitDefender), vsserv.exe (BitDefender), NortonSecurity.exe, nsWscSvc.exe (Norton), ccSvcHst.exe (Norton), MBAMService.exe (Malwarebytes), mbamtray.exe (Malwarebytes).

Notable: Avast-specific hook checks (`aswhook.dll` string present in the binary) and Kaspersky filesystem driver (`kavfs.exe`) are explicitly targeted.

Process injection capability

The binary imports and resolves the full toolkit for process injection:

- `NtCreateThreadEx`, `NtOpenProcess`, `NtWriteVirtualMemory`, `NtAllocateVirtualMemory`, `NtProtectVirtualMemory`, `NtReadVirtualMemory`
- `VirtualAllocEx`, `WriteProcessMemory`, `VirtualProtect`, `CreateProcessA`, `ResumeThread`
- `CreateToolhelp32Snapshot`, `Process32First`, `Process32Next` (process enumeration)

This is consistent with Vidar's technique of launching browsers in debug mode and injecting shellcode to steal encryption keys from process memory -- the same approach documented in

Trend Micro's Vidar 2.0 analysis.

Dead drop C2 via Telegram and Steam

Rbin emulation (125 seconds, 59 million instructions, 2,437 API calls) confirmed the live C2 protocol. The sample uses WinHTTP directly -- not Go's standard http library -- to connect to two kinds of endpoints:

Primary C2: Raw IP HTTPS to 135[.]181[.]237[.]59:443 (Hetzner, Finland). Multipart form-data POST with fields `hwid` and `build_id`.

Dead drop URLs: Telegram and Steam profile pages. The sample connects to these URLs, which historically serve as a dead-drop resolver -- the profile description or page content is used to publish the current active C2 address. This gives the operator the ability to rotate infrastructure without rebuilding the binary.

Dead drops:

- `https://telegram[.]me/hgo9tx`
- `https://steamcommunity[.]com/profiles/76561198707628078`

The request format uses multipart boundaries:

```
Content-Type: multipart/form-data; boundary=----{boundary}
Cache-Control: no-cache
```

POST body:

```
-----{boundary}
Content-Disposition: form-data; name="hwid"
{hwid}
-----{boundary}
Content-Disposition: form-data; name="build_id"
{build_id}
-----{boundary}--
```

Two User-Agent strings are used in different contexts:

- `Mozilla/5.0 (Macintosh; Intel Mac OS X 10.15; rv:140.0) Gecko/20100101 Firefox/140.0` -- used for dead drop requests
- `Mozilla/5.0 (Windows NT 10.0; Win64; x64; rv:153.0) Gecko/20100101 Firefox/153.0` -- used for C2 communication

The binary logs dead drop results with the format `"Dead drop: %s (sw: %s)"`.

Config

The extracted config from Triage's behavioral analysis:

```
version: 1.5
family: vidar
botnet: 702ef1b4007f07887e9faaee0667b50b
rule: Vidar_v1_5
c2:
  - https://telegram[.]me/hgo9tx
  - https://steamcommunity[.]com/profiles/76561198707628078
user_agent: Mozilla/5.0 (Macintosh; Intel Mac OS X 10.15; rv:140.0) Gecko/20100101 Firefox/140.0
```

IOC summary

Hashes

Type	Value
SHA256 (binary)	2995ffb73342453b258926ec865c724e3567eee1bb8eb35d61796ee0c4f25105
SHA1 (binary)	488d2dd8768e3b804179e7f0cdcebd0a7eec52b3
MD5 (binary)	87332fcdf79e1c0bfb7713e9a52c0313

Network

Type	Value	Context
IP	135[.]181[.]237[.]59:443	Vidar C2 (Hetzner, Finland)
URL	https://telegram[.]me/hgo9tx (opens in new tab)	Dead drop resolver
URL	https://steamcommunity[.]com/profiles/76561198707628078 (opens in new tab)	Dead drop resolver

Type	Value	Context
IP	149[.]154[.]167[.]99:443	Telegram infrastructure
IP	2[.]22[.]96[.]50:443	Steam CDN (Akamai)
IP	142[.]250[.]151[.]94:80	Google (time/liveness)

C2 infrastructure

Detail	Value
C2 IP	135[.]181[.]237[.]59
Provider	Hetzner Online GmbH (AS24940)
Location	Finland
Port	443 (HTTPS)
Protocol	WinHTTP, multipart form-data POST
Auth fields	hwid, build_id

Behavioural

Technique	Detail
Anti-debug	NtSetInformationThread HideFromDebugger
Sandbox evasion	12-category scoring (6/9 to pass)
Process injection	NtCreateThreadEx, NtWriteVirtualMemory, VirtualAllocEx
C2 discovery	Dead drop resolver via Telegram/Steam profile pages
Crypto	AES-NI, ChaCha20, RC4, Base64, MurmurHash3
Browser theft	CryptUnprotectData, BCrypt decryption

Technique	Detail
Language	Go 1.25.4 native PE
Build path	2808a942/46D2315

If you operate a threat intelligence platform or run Triage infrastructure and can share data, reach out. Additional sample visibility directly sharpens the tracking.

Kirk

I like the internet. Want to get in touch?