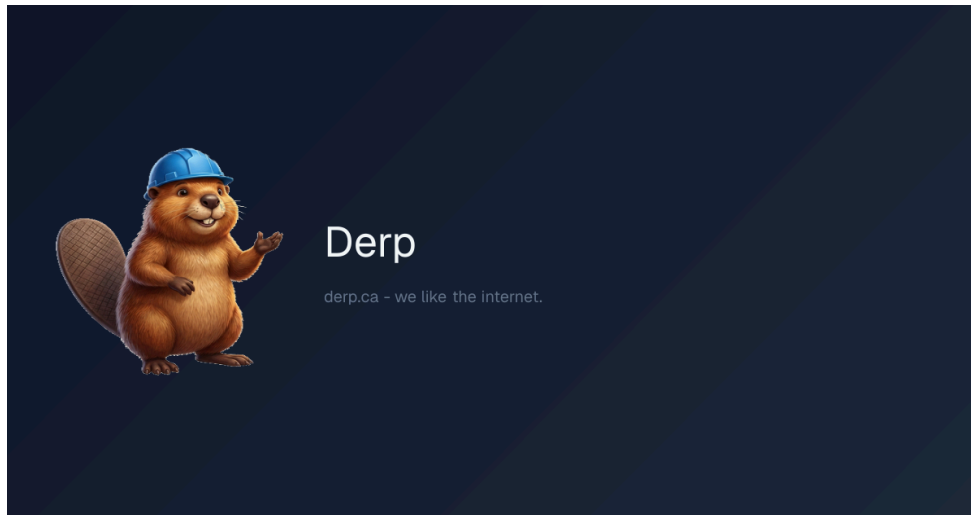


SERPENTINE#CLOUD returns: ClickFix lure drops five RATs

 derp.ca/research/serpentine-cloud-clickfix-return

Kirk

April 3, 2026



`malware clickfix brute-ratel xworm python cloudflare`

► On this page

Matching GUIDs, encryption keys, infrastructure, and tool reuse tie this activity to the same operator behind SERPENTINE#CLOUD - five weeks after remediation. Same target organisation. Same five RAT families. New delivery chain.

The previous breach - covered across five posts starting with [PureLogs](#) - ran undetected for roughly six months before it was flagged in February 2026. The attack vector was never confirmed. This time, it was ClickFix social engineering through ephemeral Cloudflare tunnels, caught at stage 0 before any payload executed.

The payload set is familiar: VenomRAT, AsyncRAT, [XWorm/Violet](#) (upgraded from v4.7 to v5), PureHVNC, and a new addition - Brute Ratel C4. BRc4 replaces the Annorii loader from the prior campaign. It does not operate its own C2. Instead, it decrypts and injects PureHVNC stage 2 into notepad.exe using Early Bird APC (asynchronous procedure call) queue injection with PPID (parent process ID) spoofing to explorer.exe.

Five families, four C2 endpoints. The BRc4 binary is an injection wrapper - direct syscalls and process mitigation policies that block non-Microsoft DLL injection into its target process.

All four C2 endpoints resolve to DuckDNS domains on two IPs in the same AT&T residential /24 in Chicago. The operator consolidated from three countries to a single subnet. Infrastructure pivoting attributed 13 DuckDNS domains to this operator across 12 historical IPs in six countries, with the earliest confirmed activity dating to July 2025.

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.

What changed

The [Python loader pipeline](#) is the same. Same Kramer bytecode obfuscator (key: `sparkle`). Same Donut v0.9.2 x64 shellcode. Same Early Bird APC injection template. Same `import base64, ctypes, sys, os` header. Same embedded hex keys structure. The template has not changed in eight months.

What changed is the wrapper around it.

Component	February 2026	March 2026
Delivery	Unknown (direct execution suspected)	ClickFix WSH lure via Cloudflare tunnels
Obfuscation	Kramer on some loaders, plaintext on others	Kramer on all 10 loaders
XWorm/Violet	v4.7 (1.4 MB)	v5 (71 KB - 95% smaller)
PureHVNC wrapper	Annorii loader	Brute Ratel C4 (PPID spoof, direct syscalls)
PureHVNC obfuscation	ConfuserEx	Custom XOR+AES + TLS cert pinning
C2 infrastructure	3 IPs across Hungary, Netherlands, US	2 IPs in same Chicago /24
DuckDNS domains	Rotated per campaign	Rotated - but 902399hfjks persisted from prior campaign

The Violet install GUID (3d847c5c-4f5a-4918-9e07-a96cea49048d), separator (xSXSXSX), VenomRAT salt (DcRatByqwqdanchun), PureHVNC ProtoBuf architecture (86 message types), and Contacts-folder staging path all match the February campaign.

Kill chain

Stage	Component	Action
0	Scan_0620954916911.pdf.wsh	rundll32 + davclnt.dll WebDAV fetch via CF tunnel #1
1	UKMar30.wsf	Copies 3 batch files from CF tunnel #2
2	UKM301.bat	Downloads 2 ZIPs + persistence script from CF tunnel #3
3	UKM302/303.bat	Runs all .py via embedded Python 3.12, hides dirs, self-deletes
4	10 Kramer .pyc payloads	AES+XOR or RC4 decrypt to Donut v0.9.2 x64 shellcode
5	Donut shellcode	AMSI/WLDP bypass, CLR bootstrap, .NET assembly load
6	Final payloads	VenomRAT, AsyncRAT, XWorm/Violet v5, PureHVNC, Brute Ratel C4

Endpoint protection detected rundll32.exe loading davclnt.dll for a WebDAV fetch to the first Cloudflare tunnel at 13:56:47 UTC on March 31. The host was isolated within seconds. No payloads reached disk.

Delivery chain (stages 0-4)

Stage 0: WSH lure

Scan_0620954916911.pdf.wsh is a 127-byte Windows Script Host settings file with a .pdf.wsh double extension. It instructs rundll32.exe to load davclnt.dll for a WebDAV/SSL fetch from the first Cloudflare tunnel. The target path is /yop/ on edward-fwd-vacuum-change.log.trycloudflare.com, which was running WsgiDAV 4.3.0 with anonymous read-write access.

Stage 1: WSF loader

UKMar30.wsf is an 869-byte JScript file. It copies three batch files from the second tunnel (handed-mines-abc-intensity.trycloudflare.com, path /ku/) and executes them sequentially with 10-second sleeps between each.

Stage 2: Batch downloader

UKM301.bat relaunches itself hidden via a VBS helper, then uses `curl` to download two ZIP archives and a persistence batch file from the third tunnel (`represents-causes-conflicts-silver.trycloudflare.com`). The ZIPs extract to hidden directories under `%USERPROFILE%\Contacts`:

- 1Mar23MA.zip (16.7 MB) -> MainRingtones
- 1Mar23ST.zip (16.6 MB) -> str
- 1Mar23SU.txt (2.7 KB) -> Startup persistence batch

Each ZIP contains Python 3.12 x64 (embedded runtime) and five Kramer-obfuscated `.py` payloads. The two sets are redundant - same families, different encryption keys.

Stage 3: Executor

UKM302.bat and UKM303.bat are identical. They execute all `.py` files via the embedded `python.exe`, hide the payload directories with `attrib +h`, kill parent processes via WMI VBS helpers, then delete themselves and the VBS files. All five RAT families launch simultaneously.

Stage 4: Persistence

1Mar23SU.txt is a batch file disguised with a `.txt` extension. It runs on boot from four paths including `%APPDATA%\Winic\30.3.0rc50` (a 32-bit Python copy) and uses `DiscordDial.vbs` as a launch helper.

Kramer obfuscator

All 10 payload files are Python 3.12 bytecode (`.pyc` magic `cb0d0d0a`) with `.py` extensions. The `Kramer` class decodes source at runtime and passes it to `eval()`.

The encoding converts source characters to CJK Unicode codepoints via a per-file offset, then hex-encodes the segments and joins them with `/` delimiters. Printable ASCII symbols and uppercase letters are shifted by +1. Newlines are encoded as Greek zeta (`U+03B6`).

Constants across all files: `key = sparkle`, `alphabet = abcdefghijklmnopqrstuvwxyz0123456789`. The CJK base offset varies per file (`0x439E` to `0xE03F0` - upper values fall outside standard CJK blocks into supplementary private-use ranges). Method names are randomised.

We decoded all 10 files natively using Python 3.12 with a hooked `eval()` call.

Python loader templates

Two templates across the 10 files:

Template 1: AES+XOR shellcode injector (8 files - Ann, Ass, Hvv, Vio x 2 sets)

Decryption: AES-256-CBC -> XOR with key2 -> XOR with key1. Each file has unique keys. Injection: `CreateProcessA(explorer.exe, CREATE_SUSPENDED)` -> `VirtualAllocEx` -> `WriteProcessMemory` -> `QueueUserAPC` -> `ResumeThread`. Early Bird APC queue injection into a suspended explorer.exe process.

Template 2: RC4 shellcode loader (2 files - 2Laz x 2 sets)

Decryption: RC4 with 8-byte ASCII key (`qgCzmCuY` for MA set, `k5m60Ubu` for ST set). Execution: `VirtualProtect(PAGE_EXECUTE_READWRITE)` -> `CFUNCTYPE` cast -> direct call. No process injection - runs shellcode in the Python process.

Both templates produce Donut v0.9.2 x64 shellcode. All six unique shellcode blobs (5 families, one shared between sets) use Chaskey-LTS encryption with the instance header at offset `0x5`. The Donut loader bypasses AMSI (Antimalware Scan Interface) and WLDP (Windows Lockdown Policy), then bootstraps the CLR (Common Language Runtime) via `CorBindToRuntime` to load the .NET payload - or executes the native PE directly for BRc4.

Final payloads

Family	SHA256	Type	Size	C2
VenomRAT v3.6	58d9f039ec38bbe03a1e1bf58a0102ce9c94d6efe39d2450cb44917d4a5c75af	.NET PE32	70,952	y57kdsa.duckdns.
AsyncRAT 0.5.7B	4bb4a303b8e4873401be1cea68d50bdaa454471685dc30ad61e9ef746181aa29	.NET PE32	52,520	uejrhnfq.duckdns.
PureHVNC loader	f56a53ec6817c918d9a0056277022d694a06727bc9064bee95e4b80c50067f2a	.NET PE32	337,704	-
PureHVNC stage 2	59079dbdfb0346deae4efc361d78844141bf77d916adec96b23d8061e20e123c	.NET DLL	788,480	bsmaopm.duckdns.
XWorm/Violet v5	8cda591f526a09954c7a60337daa767be7948367ee52accebc30061be1dc581a	.NET PE32	71,464	vivogrouplink.duckdns.
Brute Ratel C4	026f71d40fa2e3c530283c1a70925d14eeee18d98f95506dd88cb698ccca6859	PE32+ x64	621,864	shares PureHVNC

VenomRAT v3.6

VenomRAT v3.6 is a dcRAT/qwqdanchun fork. AES with passphrase **EqobtaJh1ra112Px0fjvG8Ircxdf2e2P**, salt **DcRatByqwqdanchun** (PBKDF2, 50,000 rounds). Certificate CN: **EB0LA**. AMSI and ETW bypass via memory patching. Plugin-based architecture. Install flag set to false - no persistence deployed. The process kill list targets ProcessHacker, Taskmgr, Regedit, and several Windows Defender components.

AsyncRAT 0.5.7B

AsyncRAT 0.5.7B is a lighter build of the same codebase. AES with passphrase **Ff6VygGEmXLxZ17uU1fqBwyv7Not5Jtw**. Mutex: **AsyncMutex_6SI80kPnk**. Plugin-based, install flag false. No AMSI/ETW bypass, no process kill list. Both VenomRAT and AsyncRAT perform System Language Discovery (T1614.001) by reading **HKLM\SYSTEM\ControlSet001\Control\NLS\Language**.

XWorm/Violet v5

XWorm/Violet v5 dropped from 1.4 MB (v4.7) to 71 KB. The config is double-base64 encoded then XOR'd with **mwrQqYk**. Mutex: **HOHE6S8FaZZ1Gf0f**. 80+ commands covering reverse shell, keylogger, HVNC, credential theft, DDoS, crypto clipper, USB worm propagation, ngrok tunnelling, and BSOD. The Violet install GUID matches the February campaign.

Brute Ratel C4 loader

The BRc4 binary is a 621 KB native x64 PE that impersonates mmc.exe (Microsoft Management Console, v10.0.20348.4163) via its **VS_VERSION_INFO** resources. PE timestamp: 2026-03-20. The **dnSUXRIL** marker string appears twice in the binary - a known BRc4 identification signature.

PE structure

Section	Size	Purpose
.text	176 KB	Loader code, FNV-1a API resolution
.data	370 KB	Encrypted payload blob (high entropy)
.retplne	-	Retpoline stubs

Section	Size	Purpose
_sysc	-	Direct syscall stubs

The entire import table is minimal - `IsProcessorFeaturePresent`, `SetUnhandledExceptionFilter`, and a few CRT functions. All operational APIs are resolved at runtime by walking the PEB `InLoadOrderModuleList` and hashing export names with FNV-1a (offset basis 0x811c9dc5, prime 0x1000193).

Payload decryption

The .data section contains a 16-byte header followed by 369,936 bytes of ciphertext.

Offset	Field	Value
0x00	uint64 injection_method	1 (Early Bird APC Queue)
0x08	uint64 reserved	0
0x10	AES-256-CBC encrypted blob	369,936 bytes

Decryption:

Parameter	Value
AES-256-CBC key	a984c0ca515684ae39ef9af5f3872daf30bae6b91b141258914db55bf5b5869e
AES IV	acb58f89cc0d198655d8a648df74b51f
Post-decrypt XOR	cfc1b3813d370a1c2ebcc16562d7f62b (16-byte rotating key)

The AES implementation uses standard Rijndael S-boxes in .rdata. After CBC decryption, a second pass XORs each byte with the rotating 16-byte key.

Injection

Ten injection methods are compiled into the binary, each confirmed present via static analysis of the .text section. The config byte at .data+0x00 selects the active method:

Selector	Method
1	Early Bird APC Queue (active in this sample)
2	KernelCallbackTable
3	Thread Hijacking
4	Section View Mapping
5	Thread Suspension
6	LineDDA Callback

Selector	Method
7	EnumSystemGeoID Callback
8	FLS Callback
9	SetTimer
10	Clipboard

The active method (1) spawns `C:\Windows\System32\notepad.exe` in a suspended state with PPID spoofed to `explorer.exe` via `InitializeProcThreadAttributeList + UpdateProcThreadAttribute(PROC_THREAD_ATTRIBUTE_PARENT_PROCESS)`. The loader allocates RWX memory in `notepad.exe`, writes the decrypted blob, queues it as an APC on the main thread, then resumes the thread. All memory operations use direct syscalls from the `_sysc` section - 12 syscall sites total, bypassing any userland hooks.

Before injection, `SetProcessMitigationPolicy` blocks non-Microsoft DLLs from loading in `notepad.exe`. This prevents EDR agents from injecting monitoring DLLs into the target process.

The BRc4-PureHVNC link

Behavioural analysis confirmed the decrypted 370 KB blob is PureHVNC stage 2 shellcode. The injected `notepad.exe` process beacons to `bsmaopm.duckdns.org:6757` - the PureHVNC C2. Memory dumps from `notepad.exe` contain .NET CLR heap structures consistent with PureHVNC, not native BRc4 badger regions. The PPID spoof is effective enough that process attribution shows `notepad.exe` as a child of `Explorer`, not the malware. The loader exits within three seconds of completing injection.

BRc4 is the delivery and evasion layer. PureHVNC is the payload. There is no separate BRc4 C2 channel.

PureHVNC

The PureHVNC loader (`Ygfunk1` namespace) decrypts a 311,904-byte inline array with AES-256-CBC, strips a 4-byte length prefix, GZip-decompresses the result, and reflectively loads the output via `Assembly.Load`. The entry point is `InvokeMember("ManageSegmentedDic")`.

The stage 2 DLL (`PureHVNC_Lib` / `Lhjknny` namespaces, 788 KB) implements:

- **Hidden VNC**: captures the primary display at 640x480, JPEG-compressed at quality 60, streamed to C2 as `SequentialProfile` ProtoBuf messages
- **Browser credential theft**: targets 30+ Chromium-based browsers via profile directory enumeration under `AppData`. All paths are XOR-obfuscated at runtime
- **PE/DLL injection**: loads .NET assemblies via `Assembly.Load(byte[])` or patches existing loaded assemblies. Also implements native injection via `VirtualAlloc + WriteProcessMemory` with runtime-constructed API names
- **File manager**: recursive directory listing, upload, download, and delete
- **Persistence**: copies to `%AppData%\<subfolder>\<filename>` (paths obfuscated) with registry Run key persistence

The C2 protocol layers TCP + TLS 1.0 (`SslStream`) with a custom `RemoteCertificateValidationCallback` that pins the server certificate against a cert embedded in the config blob. A 4-byte big-endian length prefix frames each ProtoBuf message. 86 concrete message types are registered via `[ProtoInclude]` directives on the `ControllableInitializer` base class.

All string constants are XOR-obfuscated against per-module integer masks loaded from an AES-CBC encrypted + Deflate-compressed embedded resource at assembly load time. Static extraction of the C2 host and port is not possible - the config requires runtime decryption.

The C2 was extracted through behavioural analysis: `bsmaopm.duckdns.org:6757`, resolving to `12.202.180.133`.

Infrastructure intelligence

C2 resolution

Family	C2	Port	IP (2026-04-03)
AsyncRAT	uejrhnfq.duckdns.org	6745	12.202.180.133
VenomRAT	y57kdsa.duckdns.org	7878	12.202.180.133
PureHVNC + BRc4	bsmaopm.duckdns.org	6757	12.202.180.133
XWorm/Violet	vivogrouplink.duckdns.org	2128	12.202.180.105

Both IPs are in 12.202.176.0/21 - AS7018, AT&T Enterprises, Chicago, Illinois. ISP-class infrastructure with no rDNS, no Shodan port history, no GreyNoise mass-scan activity, and no blacklist entries (all queried 2026-04-03).

Domain inventory

Reverse IP lookups and passive DNS (SecurityTrails, queried 2026-04-03) revealed 13 DuckDNS domains attributed to this operator:

Active on 12.202.180.133 (9 domains): bsmaopm (PureHVNC+BRc4), uejrhnfq (AsyncRAT), y57kdsa (VenomRAT), 902399hfjks (XWorm V3.1 - persisted from the February campaign), jbsak, 5840dfhk, mdata4717, hy647dhon, hjupow

Active on 12.202.180.105 (2 domains): vivogrouplink (XWorm/Violet v5), volvogroup20 (continues the vi*group* naming convention)

Sinkholed (2 domains): remgreat2740 and anongroup resolve to 192.169.69.25/.26 - HYAS sinkholes (sinkhole.hyas.com). These were formerly on ServerAstra infrastructure in Budapest used in the February campaign.

Infrastructure timeline

DNS history for 902399hfjks.duckdns.org - the domain that persisted across both campaigns - shows the migration path:

Period	IP	Hoster
Nov 2025	206.223.183.200	Beanfield Technologies, Toronto
Nov 2025 - Jan 2026	45.58.143.254	Sharktech, Amsterdam
Jan 2026 - present	12.202.180.133	AT&T, Chicago

The Sharktech period overlaps with the February campaign, which also used 45.58.143.254 for Violet, Plog, and PureHVNC C2s.

remgreat2740.duckdns.org bounced between three LeaseWeb Netherlands IPs and AT&T Chicago before HYAS sinkholed it on February 20. The consolidation to AT&T Chicago followed the HYAS sinkholing in February.

Communicating files

VirusTotal passive data on 12.202.180.133 (queried 2026-04-03) shows 10 communicating files - five from this case and five older samples from the same operator. The oldest is PN0LazJuly21_hv.py, a Python loader with the BRc4-to-PureHVNC naming pattern, hosted on jbsak.duckdns.org. This dates the operator's activity to at least July 2025.

One of the related samples - 1MaDLL.zip - contained a parallel DLL-based injection wrapper dated March 30 (the same day as this campaign). The DLL exports inject_early_bird and xor_decrypt, matching the BRc4 loader's injection method and cryptographic approach but packaged as a regsvr32-loadable DLL. The operator maintains multiple injection tools alongside the Python loader pipeline.

Obfuscation evolution

Samples from September 2025 show that Kramer obfuscation was applied selectively. In that period, only the BRc4 loader (2Sep24Laz_hv-obf.py, note the -obf suffix) was Kramer-wrapped. The standard AES+XOR loaders shipped as plaintext Python with inline hex keys. By March 2026, all 10 files are Kramer-obfuscated. The injection template itself - same imports, same injection logic, same cryptographic structure - has not changed.

The PureHVNC loader also shrunk from 3.4 MB (February 2026 build) to 338 KB (this campaign). Same Ygfumk1 namespace in both builds.

IOC summary

Network

Type	Value	Context
C2	uejrhnfq.duckdns.org:6745	AsyncRAT 0.5.7B
C2	y57kdsa.duckdns.org:7878	VenomRAT v3.6
C2	vivogrouplink.duckdns.org:2128	XWorm/Violet v5
C2	bsmaopm.duckdns.org:6757	PureHVNC + BRc4
C2 IP	12.202.180.133	AsyncRAT, VenomRAT, PureHVNC+BRc4
C2 IP	12.202.180.105	XWorm/Violet v5
Delivery	edward-fwd-vacuum-changelog.trycloudflare.com	WSH lure (/yop/)
Delivery	handed-mines-abc-intensity.trycloudflare.com	WSF loader (/ku/)
Delivery	rover-earlier-baseline-karen.trycloudflare.com	Batch files
Delivery	represents-causes-conflicts-silver.trycloudflare.com	Payload ZIPs

Hashes

SHA256	File
e84cbbbc018d7e54c5afed760f04c06731ba57c1d40414c8b94ba1c488b9c9c5	Scan_0620954916911.pdf.wsh
6b45e1a38609b9b7f2f2508b0b38f700a75ee1ea9b6c548d1a086bd91863efc3	UKMar30.wsf
e06dd348a334de7e2e43ef7a3739d4b4cf792b615595262aa212eec4e3005564	UKM301.bat
218628edc95f7c425fad294048adca65e235ae3024f084c9afaf483f66f71b6c	UKM302.bat / UKM303.bat
3bc36b9b7bc5ee73b26dd94d34a31cb707feb9a68d2e4832d276e9274e780a34	1Mar23MA.zip
010ce592bcabf0d4e786b20d46bbd25893734a176e1f5322a5f28c4f94d4c6e1	1Mar23ST.zip
58d9f039ec38bbe03a1e1bf58a0102ce9c94d6efe39d2450cb44917d4a5c75af	VenomRAT v3.6
4bb4a303b8e4873401be1cea68d50bdaa454471685dc30ad61e9ef746181aa29	AsyncRAT 0.5.7B

SHA256	File
f56a53ec6817c918d9a0056277022d694a06727bc9064bee95e4b80c50067f2a	PureHVNC loader
59079dbdfb0346deae4efc361d78844141bf77d916adec96b23d8061e20e123c	PureHVNC stage 2
8cda591f526a09954c7a60337daa767be7948367ee52accebc30061be1dc581a	XWorm/Violet v5
026f71d40fa2e3c530283c1a70925d14eeee18d98f95506dd88cb698ccca6859	Brute Ratel C4

Host

Indicator	Context
%USERPROFILE%\Contacts\MainRingtones	Primary payload directory (hidden)
%USERPROFILE%\Contacts\str	Secondary payload directory (hidden)
%APPDATA%\Winic\30.3.0rc50	32-bit Python persistence
AsyncMutex_6SI80kPnk	AsyncRAT mutex
H0HE6S8FaZZlGf0f	XWorm/Violet mutex
WinSc32.exe	XWorm drop filename (%TEMP%)

Behavioural

Indicator	Context
rundll32.exe + davclnt.dll WebDAV to trycloudflare.com	Stage 0 delivery
notepad.exe with PPID = explorer.exe, beaconing to DuckDNS	BRc4 -> PureHVNC injection
NtCreateUserProcessBlockNonMicrosoftBinary on spawned process	BRc4 mitigation policy
DNS queries to DuckDNS domains every 12-17 seconds	C2 retry loops (all families)

Campaign assessment

The operator retargeted five weeks after remediation with an upgraded delivery chain - ClickFix social engineering and ephemeral Cloudflare tunnels - and a new evasion layer in Brute Ratel C4. The underlying toolchain has been stable since at least July 2025: same Python loaders, same Donut shellcode, same five RAT families, same DuckDNS infrastructure pattern. The infrastructure consolidation to a single AT&T /24 in Chicago followed the sinkholing of their European infrastructure.

See also: [Python loader evolution](#), [Violet RAT](#), [PureLogs RAT](#), [PureCrypter](#), [Remcos Autolt](#).

Kirk

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